DATA POOL ARCHITECTURE, SYSTEM, AND METHOD FOR INTELLIGENT OBJECT DATA IN HETEROGENEOUS DATA ENVIRONMENTS

Inventors:
Robert Allen Stanley
Erich Alfred Gombocz

RELATED APPICATIONS

Priority is hereby claimed under 35 U.S.C. 120 and/or 35 U.S.C. 119(e) to the following United States Provisional and Utility Patent Applications, each of which is hereby incorporated by reference:

U.S. Utility Patent Application Serial No. __/___,___ (Attorney Docket No. A-70134/RMA) filed 06 December 2001 and entitled *Data Pool Architecture, System, And Method For Intelligent Object Data In Heterogeneous Data Environments;*

U.S. Utility Patent Application Serial No. __/___,___ (Attorney Docket No. A-70135/RMA) filed 06 December 2001 and entitled *Intelligent Molecular Object Data Structure and Method for Application in Heterogeneous Data Environments with High Data Density and Dynamic Application Needs*;

U.S. Utility Patent Application Serial No. __/___,___ (Attorney Docket No. A-70136/RMA) filed 06 December 2001 and entitled *Intelligent Object Handling Device and Method for Intelligent Object Data in Heterogeneous Data Environments with High Data Density and Dynamic Application Needs*;

U.S. Utility Patent Application Serial No. __/___,___ (Attorney Docket No. A-70310/RMA) filed 06 December 2001 and entitled *System, Method, Software Architecture, And Business Model For An Intelligent Object Based Information Technology Platform*;

U.S. Provisional Application Serial No. 60/254,063 filed 12/06/00 entitled *Data Pool Architecture for Intelligent Molecular Object Data in Heterogeneous Data Environments with High Data Density and Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/254,062 filed 12/06/00 entitled *Intelligent Molecular Object Data for Heterogeneous Data Environments with High Data Density and Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/254,064 filed 12/06/00 entitled *Handling Device for Intelligent Molecular Object Data in Heterogeneous Data Environments with High Data Density and Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/259,050 filed 12/29/00 entitled *Object State Engine for Intelligent Molecular Object Data Technology*;

U.S. Provisional Application Serial No. 60/264,238 filed 01/25/01 entitled *Object Translation Engine Interface For Intelligent Molecular Object Data*;

U.S. Provisional Application Serial No. 60/266,957 filed 02/06/01 entitled *System, Method, Software Architecture and Business Model for an Intelligent Molecular Object Based Information Technology Platform;*

U.S. Provisional Application Serial No. 60/276,711 filed 03/16/01 entitled *Application Translation Interface For Intelligent Molecular Object Data In Heterogeneous Data Environments With Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/282,656 filed 04/09/01 entitled *Result Generation Interface For Intelligent Molecular Object Data In Heterogeneous Data Environments With Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/282,658 filed 04/09/01 entitled *Knowledge Extraction Engine For Intelligent Object Data In Heterogeneous Data Environments With Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/282,654 filed 04/09/01 entitled *Result Aggregation Engine For Intelligent Object Data In Heterogeneous Data Environments With Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/282,657 filed 04/09/01 entitled *Automated Applications Assembly Within Intelligent Object Data Architecture For Heterogeneous Data Environments With Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/282,655 filed 04/09/01 entitled *System, Method And Business Model For Productivity In Heterogeneous Data Environments*;

U.S. Provisional Application Serial No. 60/282,979 filed 04/10/01 entitled *Legacy Synchronization Interface For Intelligent Molecular Object Data In Heterogeneous Data Environments With Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/282,989 filed 04/10/01 entitled *Object Query Interface For Intelligent Molecular Object Data In Heterogeneous Data Environments With Dynamic Application Needs*;

U.S. Provisional Application Serial No. 60/282,991 filed 04/10/01 entitled *Distributed Learning Engine For Intelligent Molecular Object Data In Heterogeneous Data Environments With Dynamic Application Needs*; and

U.S. Provisional Application Serial No. 60/282,990 filed 04/10/01 entitled *Object Normalization For Intelligent Molecular Object Data In Heterogeneous Data Environments With Dynamic Application Needs*;

each of which U.S. utility and U.S. provisional patent application is hereby incorporated by reference in its entirety.

Field of Invention

This invention pertains generally to system, method, computer program product, data structure and architecture, data management, and software architecture; and more particularly to system, method, computer program product, and data structure and architecture, data management, and software architecture in the life sciences, biotechnology, therapeutic diagnostic and intervention, pharmaceuticals, and bioinformatics, also extendable to other scientific, business and information-oriented application domains.

BACKGROUND

As demand for effective Information Technology (IT) software to provide global data access and integrated scientific and business solutions has grown, significant challenges have become evident. A central problem poses access, integration, and utilization of large amounts of new and valuable information generated in each of the major industries. Lack of unified, global, real-time or near-real-time data access and analysis is detrimental to crucial business processes, which include new product discovery, product development, decision-making, product testing and validation, and product time-to-market. Additionally, the importance of functionally integrating multiple dimensions of heterogeneous data in the field, such as protein expression

data, chemical structure data, bioassay data and clinical text data, is recognized. (See for example, Lin, D., et al.: **American Genomic/Proteomic Technology** (2001) 1 (1): 38-46.)

With the completion of the sequence of the human genome and the continued effort in understanding protein expression in the life sciences, a wealth of new genes are being discovered that will have potential as targets for therapeutic intervention. As a result of this new information, however, Biotechnology and Pharmaceutical companies are drowning in a flood of data. In the Life Sciences alone, approximately one Terabyte of data is generated per company and day, of which currently the vast majority is unutilized for several reasons.

Data are contained in diversified system environments using different formats, heterogeneous databases and have been analyzed using different applications. These applications may each apply different processing to those data. Competitive software, based on proprietary platforms for network and applications analysis, have utilized data platform technologies such as SQL with open database connectivity (ODBC), component object model (COM), Object Linking and Embedding (OLE) and/or proprietary applications for analysis as evidenced in issued patents, such as for example patents issued or assigned to such companies as Sybase, Kodak, IBM, and Cellomics in US 6161148, US 6132969, US 5989835, and US 5784294, for data management and analysis, each of which patents are hereby incorporated by reference.

Because of this diversity, despite the fact that the seamless integration of public, legacy and new data is fundamentally important and/or crucial to efficient drug discovery and life science research, current data mining tools, methodologies, and technologies cannot handle all data and analyze their functional relationships simultaneously or in an acceptable time frame. There is a significant lack of data handling methods, particularly data handling methods that can utilize these data in a secure, manageable way. Security is important in at least many situations so as to prevent other competitive investigators from taking advantage of another's investigative strategies, results or queries. The shortcomings of these conventional tools, methodologies, and technologies are evident within heterogeneous software and hardware environments with global data resources. Despite the fact that the seamless integration of public, legacy and new data is crucial to efficient research (particularly in the life sciences), product discovery (such as for example drug, or treatment regime discovery) and distribution, current data mining tools, methods, and technologies cannot handle or validate all (or even a reasonable portion of) diverse data simultaneously or within an acceptable time frame.

With the expansion of high numbers of dense data in a global environment, user queries often require costly massive parallel or other supercomputer-oriented processing in the form of

mainframe computers and/or cluster servers with various types of network integration software pieced together for translation and access functionality as evidenced by such companies as NetGenics, IBM and ChannelPoint (see for example US 6125383, US 6078924, US 6141660, and US 6148298, each of which is incorporated by reference herein); packaged in some manner (e.g. Java, CORBA, "wrapping", XML) and networked supercomputing hardware as evidenced by such companies as IBM, Compaq and others in issued patents (see for example, US 6041398, and US 5842031, each of which is incorporated by reference herein). Even with these expensive software and hardware infrastructures, significant time-delays in result generation remain the norm.

In part due to the flood of data and for other reasons as well, there is typically a significant redundancy within the data, making queries more time consuming and less efficient in their results. Tools are not yet in place which can effectively detect data redundancy over heterogeneous data types and network environments, especially of data content subsets within data files, and provide ranked and validated multiple addressing and/or flagging and/or removal of such redundant data. The flood of new and legacy data results in a significant redundancy within the data making queries more time consuming and less efficient in their results.

With the advent of distinct differentiations in the field of genomics, proteomics, bioinformatics, and other information intensive fields, and the need for informed decision making in the life sciences and other disciplines, the state of object data is crucial for their overall validation and weight in complex, multi-disciplinary queries. This is even more important due to inter-dependencies of a variety of data at different states. Furthermore, because biological data frequently describe a "snapshot" representing a unique moment in time of complex processes at a defined state of the organism, data obtained at any time refer to this unique phase of metabolism. Thus, in order to account for meaningful comparison, only data in similar states can or should be utilized. Therefore, there is a growing need for an object data state processing engine, which allows one to continuously monitor, govern, validate and update the data state based on any activities of data or data objects in real-time or near-real-time, or even within an acceptable time frame. Currently, these capabilities are not broadly available for network data structures, and they are not available for data structures integrating heterogeneous data over distributed network environments.

Furthermore, these differentiations in the field of genomics, proteomics, bioinformatics, and other fields and the need for informed decision making in the life sciences, access to all data is crucial for overall validation and weight in complex, multi-disciplinary queries. This is even

more important due to inter-dependencies of a variety of data at different states. The current individual data translation approach does not support these needs. Most of these problems require real-time processing; automated, instant data translation of data from different sources; and integration of heterogeneous applications and analytical components for their solutions. Data contained in diversified system environments may use different formats, heterogeneous databases and different applications, each of which may apply different processing to those data. Therefore, there is a growing and unmet need for an automated object data translation methodologies and/or systems and structures or engines, which allows for bi-directional translation of multidimensional data from various sources into intelligent molecular objects in real-time. Currently, data translation processes between different data types are time-consuming and require administrative exchange of information on data structures application programming interfaces (API's) and other dependencies. For example, some of the latest technologies such as Incellico's CELL, IBM's DiscoveryLink, Netgenic's Synergy and Tripos' MetaLayer solutions (see Haas et al 2001) have such requirements. These processes, although available and used, have a number of shortcomings. Despite the fact that the rapid seamless integration of public, legacy and newly emerging data is crucial to efficient drug discovery and life science research, unique "wrappers" or translation layers must currently be designed and programmed in order to translate each of those data sets correctly, and even with this manual integration, multiple data types and dimensions of data interdependencies are not made available, or "functionally integrated" for detailed qualitative and quantitative comparison and analysis across data types and dimensions. These solutions currently require significant effort and resources in both, software development and data processing, and the need for improvements such as those offered by this invention are recognized.

An additional consideration, which is prohibitive to change towards a more homogeneous infrastructure is the missing of fluently definable object representation definition protocols to prepare and present data objects for unified, functionally integrated interaction within heterogeneous environments. There is a lack of defined sets of user interaction and environment definition protocols needed to provide means for intelligent data mining and optimization of multidimensional analysis to achieve validated solutions. Data currently are interacted with and presented in diverse user interfaces with dedicated, unique features and protocols, preventing universal, unified user access. Thus, a homogeneous, unified presentation such as a flexibly network-enabled graphical user interface, which integrates components from diverse applications and laboratory systems environments over a variety of connections and protocols, is highly

desirable, but heretofore non-existent for real-time data access and analysis utilizing diverse applications and data.

Finally, an additional consideration, which is prohibitive to change towards a homogenous data and applications infrastructure, is cost. The cost to bring legacy systems up to date, to retool a company's intranet-based software systems, to create a unified environment utilizing existing software products and tools such as CORBA, JAVA, XML, SQL and classic data warehousing techniques, can be time-consuming and expensive. Conventional practices require retooling and/or translating at both application and hardware layers, as evidenced by the efforts of such companies as Unisys and IBM (See for example, US 6038393 and US 5634015), and may be prohibitively expensive for smaller and medium-sized companies or groups wishing to access this type of functionality.

Because of the constraints outlined above, it is nearly impossible to extract useful, functionally integrated information from the entity of data within reasonable computing time and efforts. For these reasons, the development of comprising a unique architecture and system, comprising a unique application framework, data structure, and database structure, is unavailable and needed to overcome these obstacles (see for example, Hobbs, D.W., **Chemical and Engineering News**, (2001) <u>79</u> (13): 266, incorporated by reference herein).

Therefore, there remains a need for a system, method, computer program and computer program product, data structure, architecture, and application framework that overcome these problems and limitations.

<u>LITERATURE</u>

Andreoli, J-M., In: Agha G., Wegner P., Yonezawa A.(eds.): **Research Directions in Concurrent Object-Oriented Programming,** MIT Press (1993): 260-263; Bertino E., Urban S., Rundensteiner E.A.(eds.): **Theory and Practice of Object Systems** (1999) <u>5</u> (3): 125-197; Chaudhri A.B., McCann J.A., Osmon P.: **Theory and Practice of Object Systems** (1999) <u>5</u> (4): 263-279; Cai D. , McTear M.F., McClean S.I.: **International Journal of Intelligent Systems** (2000): <u>15</u> (8): 745-761; Hert C.A., Jacob E.K., Dawson P.: **Journal of the American Society for Information Science** (2000) <u>51</u> (11): 971-988; Hobbs, D.W., **Chemical and Engineering News**. (2001) <u>79</u> (13): 266; Lin, D., et al.: **American Genomic/Proteomic Technology** (2001) <u>1</u> (1): 38-46; Williams R. J., In: Miller W., Thomas I., Sutton R. S., Werbos, P. J. (eds.): **Neural Networks for Control**, MIT Press (1990): 97-114; Wilson G. V., Lu P. (eds.): **Parallel Programming and C++,** MIT Press (1996): 257-280.; C. N. Lauro, G. Giordano, R. Verde:

Applied Stochastic Models and Data Analysis: A multidimensional approach to conjoint analysis (1998) 14 (4): 265-274; Meyer, Bertrand: **IEEE Computer.** (1999) **32** (1): 139-140.; Chalmers, Mathew: Journal of the American Society for Information Science. (1999) **50** (12): 1108-1118.; Teasley, Stephanie and Steven Wolinsky: **Science.** (2001, June 22) **292**:2254; Haas, L.M., et al: **IBM Systems Journal.** (2001) 40 (2): 489-511.; Siepel, A., et al: **IBM Systems Journal.** (2001) 40 (2): 570-591; and Steiner, S. and Witzmann, F.A.. **Electrophoresis.** (2000) **21**: 2099-2104; Haas, L.M., et al: IBM Systems Journal. (2001) 40 (2): 489-511; each of which publications are incorporated by reference.

The following United States Patents: US 5596744, US 5867799, US 5745895, US 6076088, US 5706453, US 5767854, US 6035300, US 6145009, US 5664066, US 5862325, US 6016495, US 6119126, US 6088717, US 6052722, US 6064382, US 6134581, and US 6146027; each of which publications are incorporated by reference.

SUMMARY

The invention includes multiple system, structure, apparatus, methodological, computer software/firmware, and business or operating model elements. In one preferred embodiment, these multiple and diverse elements are synergistically combined into a commercial implementation; however, it will be understood that each of the elements and/or aspects of the invention may be used separately or in anyone of the possible combinations to provide utility.

In one of these embodiments, the invention provides a novel data structure referred to as an intelligent molecular object (IMO) or for the sake of brevity an intelligent object. The phrases intelligent molecular object and intelligent object are used interchangeably in this description. Various embodiments of these intelligent molecular objects (IMO) or intelligent objects are described and may be used alone or in the context of other aspects of the invention. One particularly advantageous embodiment of the IMO includes both procedural (or active) and non-procedural (non-active) components, where the non-procedural component has features in common with (though not the same as) "data" in the conventional sense and the procedural component couples additional information with the "data" and adds functionality and capabilities to a system or method utilizing the "data." The manner in which the non-procedural component of the IMO differs from conventional data, and the manner in which the IMO is a data structure but not a data structure in the conventional sense are developed elsewhere in this description. The IMO data structure (IMO) may be a local data structure or a distributed data structure, though in its most advantageous and functional form it will be a distributed data structure.

In light of the above, it will be appreciated that in one aspect, the invention provides an intelligent object data structure (IMO) that comprises one or more of components, interfaces, structures, definitions, and methods that takes any data content (such as raw data or data including but not limited to data coupled or associated with any valid type or structure) and turns the data content into intelligent data. The intelligent data is thereby made substantially or entirely capable of any one or combination of the following: self-organizing, self-translating, capable of being processed in a parallel manner, tracked, ranked according to validation, and other features and capabilities as are described elsewhere in this description. The intelligent data is also capable of being functionally integrated, such as by mapping of data content, extraction of meta-data for each individual data object, and direct linking between objects and objects and applications.

In another one of these embodiments, the invention provides a novel applications handling system, architecture, and framework referred to as the intelligent object handler (IOH) . The IOH may be analogized to a super-browser, that functionally integrates and validates (ranks or flags according to validation status) homogeneous and heterogeneous applications and data within a unified, globally accessible framework over a variety of networks, network protocols, client or access point hardware and operating systems, and network systems (such as peer-to-peer, client-server, ASP, and other system architecture and topologies as are known in the art). In essence the IOH is hardware, operating system, application program, and network protocol neutral or agnostic. In one embodiment, the IOH applications handling framework comprises processing components, interfaces, definitions, algorithms and methods for functionally integrating and (optionally but desirably) validating diverse applications and data. As with the IMO (and the IOP described hereinafter) the IOH may be localized or distributed.

In a third primary aspect, the invention provides embodiments of intelligent object pools (IOP) which comprise a second database structure in addition to the afore described IMO data structure. Embodiments of the IOP are like a single global database of whatever data a user or other entity is interested in, that has been made accessible and compatible for analysis. (The ability to perform meaningful analysis may generally depend on the existence of meaningful relationships within the data and/or analysis applications themselves.) The IOP creates a unified, dynamically updated, data pool out of previously incompatible, possibly distributed data, and provides viewing and interactivity according to multiple, dynamically defined data content and property dimensions. Embodiments of the IPO may be locally configured or distributed, and may generally comprise one, more than one, or all of processing components, interfaces, and their comprised structures, definitions, algorithms and methods.

Each of the IMO, IOH, and IOP may typically be implemented in computer program software having a set of procedures or functional modules implemented as computer program instructions. These computer software programs executing on either general purpose computers or specialized processors having at least a processing unit and memory form systems that provide connectivity providing a user with access to data and other information via known internet and intranet infrastructures.

While each of the IMO, IOP, and IOH advantageously support the distributed and heterogeneous integration capabilities described and provide particular advantages, non-distributed (e.g. local) and non-heterogeneous (e.g. homogeneous) operation are clearly supported and provide significant improvements over conventional systems and methods.

In one particular aspect, the invention provides an intelligent object (alternatively referred to as an intelligent Molecular Object or IMO), which includes data structure and methodological and procedural components and in at least one embodiment is advantageously enabled in or as computer program software. When implemented as computer program software executing on or within processor and memory of one or more general purpose computers, the inventive data structure, database structure, and methodological and procedural components provide a system for data, access, manipulation, and analysis.

At one level, the invention provides system, structure, method, and computer program and computer program product for data and user identification and status management; functional integration of, and access to, potentially diverse data over a variety of computing infrastructures; integration of multiple data types and data dimensions for efficient and accurate multidimensional, parallel queries and analyses for diversified data content and dynamic applications needs in heterogeneous local and/or networked computing and applications environments with particular benefits and applicability to life sciences computing and applications environments.

The data and user identification and status management procedures and/or capabilities include: unified presentation and global unique identification of the Intelligent Object; identification, authentication and logging of users, sessions, and data content activity locally and/or over networks; dynamic routing of root object data, meta-data and data content locally and/or over networks; and data status, data integrity and validation state alerting and management.

The functional integration of, and access to, potentially diverse data over a variety of computing infrastructures procedures and/or capabilities include: functional content and attribute

definition for database and application access and routing; automated raw data matrix and matrix structure definition; automated translation of multiple data types and dimensions for unified processing and analysis; and functional integration of multiple data types and dimensions for unified processing and analysis.

The integration of multiple data types and dimensions for efficient and accurate multidimensional, parallel queries and analyses includes procedures and/or capabilities for meta-data indexing and query optimization; direct data-to-data information exchange; vectorized exchange of information subsets; data-enabled parallel processing; non-destructive cache-based processing; graphical data preview and detailed viewing; automated functional integration and launching of applications and activation of data related to Intelligent Object data content; automated assembly of applications and components for viewing and/or analysis relevant to specified data types and contents; and custom text annotation, linking and embedding of existing text. Each of these capabilities and/or procedures may be applied to diversified data content and dynamic applications needs in homogeneous and/or heterogeneous local and/or networked computing and applications environments, and find particular applicability and benefits to fife sciences computing and applications environments.

The invention also provides methods and procedures for the creation and modification of Intelligent Molecular Objects (or for simplification "Intelligent Objects" or even more simply "Objects") are provided which, upon user initiation, queries, data acquisition protocols or data import requests, invoke the unique object identifier property pane through a unified functional presentation layer. The unique object identifier property pane assigns each new data object a globally unique identification upon creation and generates a minimum set of functional property panes within the object, which account for unified viewing and processing. Once the object's state recording is started, active identification for all connections to and activities on the Intelligent Object are listed within the unique object identifier property pane, containing a real-time record of the entries. Methods and procedures for user and session authentication, permission or denial for data access, security and ownership management, highly selective data access and routing, Intelligent Object handling and storage are immediately provided.

An object root router component defines the origin of the object within the network, directs storage of the object within the database and reports the location of the object to the unique object identifier property pane.

An interactive content routing component defines where data content is located and where query-relevant content and/or results will be directed within the network for analysis or

presentation and reports the location of the data content to the unique object identifier property pane.

A status management component provides methods for data status validation, logging, use-tracking, auditing, synchronization, rollback enabled by said command history and non-destructive vector processing, and other state management and alerting protocols. The status management component communicates with an external object state engine component to monitor data integrity and to record the command history according to G*P-compliant LIMS requirements (such as, for example, GLP, GCP, GMP, ISO-9001, CDER, CFIR) within the object state engine (OSE) property pane, where the information is updated and provided for real-time or substantially real-time viewing. This information includes one or more of detailed activity logging, such as data acquisition state, calibration information, applied transformation or analysis processes, local and remote access attempts, access permission and denial, data integrity alerts, ranking status, and regulatory validation states.

A raw data matrix property pane within the Intelligent Object provides an overview of the full raw data content subset including content attribute information, source location, data type and comments regarding data content referred to by the Intelligent Object, regardless of originating data type or structure.

A matrix structure descriptor component provides methods for data field mapping of heterogeneous raw data to govern access to individual data subsets (byte-level workspace assignment) and to enable direct vectorized access to individual data fields.

A meta-data index property pane within the Intelligent Object provides a viewer for automatically generated or user-defined index information and brief meta-descriptions ("data-about-data") such as, but not limited to, specific data functionality or relationships to other data or data inter-dependencies based upon multi-parametric clustering, queries or application of certain analytical tools or a combination thereof to the data. This pane utilizes a meta-data index interface to communicate with external processing engines to create an index of descriptive data information and to provide this meta-data to the object pane descriptor component, which integrates relevant pane information for access and presentation. Additionally, the meta-data index is used to aggregate and integrate results of clustering and/or other data analyses and provides sets of rules to optimize access and routing based upon dynamically established query relationship trees regarding specific data functionality and/or meta-data description. The meta-data index is also used to rank parameters such as data quality, validation state, significance,

recency and accessibility to enable optimized access and routing based on data type, topic and content attributes to predefine analytical queries.

An object pane descriptor component compiles an overview of the Intelligent Object property pane characteristics to provide functional content and attribute definitions to access and route data content and applications. Said object pane descriptor component exchanges the information with components and access interfaces to provide definitions required for dynamic addressing, functional linking and vectorized access and routing of data content and processing results.

The application translator link interacts with an external data and applications handler, such as the Intelligent Object Handler, and an object access manager component to provide a dynamic list and interactive overview of applications, application components and data resources. The application translation link enables interactive linking and integration of the applications, applications components and data resources, according to user requirements and available resources.

The object access manager determines relevant property panes and selectively directs their content for functional presentation and access within a given application or database environment. Additionally, the object access manager interacts with external object translation engines to detect, define and address required and/or available data sources and to direct access and routing requests for specific data content to linked applications and/or databases and to functionally integrate data content with a variety of applications. The object access manager also provides content-attribute based algorithms to enable applications integration and inter-application communication.

An object query interface routes results of Boolean comparisons and other algorithms applied to content attributes according to its object pane descriptor relationships. This component also passes aggregated results from object-to-object direct information interchange to an external result aggregation engine for further processing and relays significant query outcomes back to the object pane descriptor property pane for presentation to the user.

An object graph preview property pane is included as limited resolution image/graphics viewer for quick graphical data review of Intelligent Objects, and additionally provides linking to detailed viewing as well as launching of content-specific analysis tools. The object graph preview property pane also includes an object graph preview processing component, which accounts for generation of such a limited resolution image/graphics (sometimes referred to as "thumbnail"

images or graphics) from non-graphical raw data content and which passes the image back to the object graph preview property pane.

An optional text annotation property pane within the Intelligent Object provides a location for customized text annotations, referencing, definitions and integration of links to external textual resources. An optional text annotation Interface, which links external components or applications such as text editors to allow for customization, formatting, reviewing and processing of the information through external editors and which allows to pass this information back to the text annotation property pane and to provide integrated support for text mining algorithms utilizing external distributed learning and/or knowledge extraction engines.

In another particular aspect, the invention provides an intelligent object handler (IOH), which includes methodological and procedural elements and in at least one embodiment is advantageously enabled in or as computer program software. When implemented as computer program software executing on or within processor and memory of one or more general purpose computers, the methodological and procedural elements provide or support a system for handling data. Methods and procedures are provided for numerous data handling tasks in both homogeneous and heterogeneous data environments, particularly those with high data density and dynamic application needs. The data handling tasks include, but are not limited to: generation of Intelligent Object data; unified presentation of dynamically customizable functional menus and interfaces such as for user definition, administration and security protocols; secured user interaction, access and presentation based on imported and/or defined user definition, administration and security protocols; data object standardization and normalization; definition of user interaction and computing environment protocols required for data object translation, standardization, access and routing; definition of data type access, translation, presentation and routing protocols for functional data and applications integration; definition of application and/or application components and interface access, translation, presentation and routing protocols for functional data and applications integration; and provision of interactive, unified, functionality for data acquisition, management, viewing and analysis.

In the object creation method and procedure, the Intelligent Object generator extracts relevant data information, routes real-time data from ongoing data acquisitions and transforms device outputs and heterogeneous data types to Intelligent Object data. Data content may be stored remotely from the corresponding Intelligent Object, and both, data content as well as Intelligent Objects (stored in "Intelligent Object Pools" or Pool subset "iPools") may be stored locally or may be distributed across heterogeneous data storage resources and networks.

Components such as the object standardization technique and the object normalization engine standardize and normalize the data by calibration according to standardized empirical criteria.

In the interactive user access and presentation method and procedures, the unified presentation layer provides the web-enabled graphical user interface that integrates the technology defined to unify diverse applications, laboratory systems environments, and Intelligent Object data at the graphic user interface layer. As an example, in the security and access methods, the user menu activates the user definition and administration shell and prompts for user input regarding access privileges environments at login. The master query component then presents security and access protocols to the unified presentation layer and to the object state engine for authentication and permits or denies access to begin fielding user queries and commands for data acquisition, retrieval, or analysis.

In the method and procedures for interactive, functionally integrated data acquisition, management, viewing and analysis, user interactivity at the front end is enabled by the unified presentation layer, which is linked to defined processing components and access interfaces.

In the environment definition method and procedures, the application and database definition generator interface dynamically detects application and database requirements and defines the computing environment for the data type translator, the application translation interface, and the application framework.

In the object definition method and procedures, the data type translator defines the data type dependencies for the Intelligent Object generator, the object translation engine and the application framework component according to the applications and database environment defined by the application / database definition generator.

At the same time, in the functional integration method and procedures, the Intelligent Object application framework provides functional integration of components, access interfaces and Intelligent Objects of the Intelligent Object Handler, to provide fast, efficient, functionally integrated querying, viewing and analysis. Components and interfaces such as the application and database definition generator interface and the application translation interface provide access and routing protocols to heterogeneous applications and databases.

Additionally, in the method and procedures for functional integration, the Intelligent Object Handler (IOH) enables activities including for example, real-time or near-real-time acquisition, management, viewing and analysis of Intelligent Object data through the utilization of integrated meta-data tags and pointers activated by the master query component and returned via components and access interfaces to the master query interface for presentation to the user.

Automated and/or user-directed interaction with external applications, processing components, instruments and devices is enabled by access interfaces including the master query interface, direct instrument acquisition and control, legacy synchronization interface, and report generation interface.

In a further particular aspect, the invention provides an intelligent object pool (IOP). Methods, procedures, and structures are provided for storage, management, processing and viewing of **Intelligent Molecular Objects** (IMO) **200** (also referred to as Intelligent Object or simply Objects) data within defined iPool subsets. The methods and processes include Pool Boundary Protocol definitions, Meta-data Query Definitions, and Pool Content Access definitions.

The Pool Boundary Protocols enable utility including security authentication, availability and access management, data exchange, data and data set integrity assessment. In an exemplary embodiment of the Pool Boundary Protocol methods, the iPool Security Authentication module (iPSAc/iPSAi) authenticates iPool data requests according to user login and Intelligent Object data and/or iPool identification. Next, the iPool Availability Monitor module (iPAMc/iPAMi) and iPool Exchange Protocol module (iPEP/iPEPi) presents iPool relationships and availability to the authenticated user. At the same time, the object integrity assessment module (OIAc/OIAi) assesses Intelligent Object integrity for security and quality assurance/quality control and the iPool Integrity Assessment module (iPIAc/iPIAi) assesses data integrity within defined **iPools 155** for security and quality assurance/quality control.

In an exemplary embodiment of the Meta-data Query methods, the Real-time Meta-data Link module (RML) comprising a Real-time Meta-data link component (RMLc) and a Real-time Meta-data link interface (RMLi) provides for rapid relevant data access based on direct linking of query parameters to Intelligent Object and iPool Meta-data Index (MDX) content. Simultaneously, the Intelligent Object-to-Intelligent Object (Object-to-Object) Query module (OQMc/OQMi) provides for rapid query optimization based on Intelligent Object and iPool intercommunication regarding query parameters and Meta-data Index (MDX) content. Depending on query parameters, an iPool-to-iPool Query module (PPQc/PPQi) may be called, which provides for query optimization based on iPool data intercommunication regarding query parameters and Meta-data Index (MDX) content contained within more than one iPool.

In the Pool Content Access methods, the aggregate Real-time Significance Generator module (aRSGc/aRSGi) provides for significance detection based on provided definitions, equations and results; interactive aggregation of reported Intelligent Object attributes within the

iPool; and on such as the correspondence of the attributes to queries, processing parameters/or other automated and/or user directed parameters. Next, the aggregate Meta-data Index generator module (aMDXc/aMDXi) provides for meta-data index generation for iPool sets of Intelligent Object meta-data, of values defining aspects of the global Intelligent Object Pool or accessed **iPools 155** based on query parameters; data content, function and relationships; and Meta-data Index (MDX) attribute content. The comprised Result Aggregation Engine module component and interface (RAEc/RAEi) enables results aggregation based on Intelligent Object Meta-data and aggregate Meta-data information and data values, and properties generated as results of automated and/or user-directed queries, commands and applied processes and methods. The Result Aggregation module provides information required for Report Generation, Meta-data Index updating and the like.

An object property-selective pre-sorting and viewing module (IMO-Zc/IMO-Zi), "IMO Zoom" which organizes Intelligent Objects based on their relationships within individual **iPools 155**, to allow for real-time exclusion of irrelevant object data layers and Intelligent Object relationship viewing.

BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 shows diagrammatic representations of two embodiments of an intelligent molecular object Information Technology (IMO IT) Platform architecture and topology, depicting structural, methodological, and functional relationships between framework, engines, interfaces, and other components, where FIG. 1A shows a first embodiment of the overall architecture and FIG. 1B shows an alternative exemplary embodiment of the IMO IT architecture having different localizations of components and interfaces, and focusing on the integrated data and applications handling aspect of the IMO IT Platform.

FIG. 2 is a diagrammatic representation of functional relationships between instruments, applications, and intelligent object pool (IOP) 204 and diverse databases within an embodiment of the intelligent molecular object (IMO) architecture and topology.

FIG. 3 is a diagrammatic representation of a typical multi-user network collaborative research effort involving geographically diverse data acquisition and query sites.

FIG. 4 is a diagrammatic representation in the form of a procedure or flow chart showing typical data processing procedures and flows within the IMO IT architecture to provide a relevant real-time or near-real-time answer to a query.

FIG. 5 is a diagrammatic representation of an exemplary system architecture or hardware configuration for the IMO(s), IOP, and IOH and its enabling architecture.

FIG. 6 shows diagrammatic representations of several embodiments of an intelligent object or intelligent molecular object (IMO) 200, where FIG. 6A and FIG. 6B show alternative exemplary embodiments of the IMO IT architecture having different views and localizations of components and engines.

FIG. 7 shows diagrammatic representations of several embodiments of an intelligent object pool (IOP) 204, where FIG. 7A shows relationships and interactivity between the intelligent object pool (IOP) and its components and access interfaces, to the intelligent objects (IMOs) 200 provided by the IOP, and to an external intelligent object handler (IOH) 202, and where FIG. 7B shows an intelligent object pool (IOP) and a data subset intra-Pool (iPool) 155.

FIG. 8 shows a diagrammatic representation of an embodiment of an object handler or intelligent object handler (IOH).

FIG. 9 is a diagrammatic illustration in the form of a chart showing an embodiment of distributed data resources, a Unified Client data subset intra-Pool (iPool) 155, access, and exchange relationships to the data resources, according to aspects of one embodiment of the invention.

FIG. 10 is a diagrammatic illustration depicting a representation of an embodiment of a process model for an exemplary embodiment of the intelligent object pool (IOP) 204, including relationships between legacy data, access protocols, and unified object content and meta-data interaction.

FIG. 11 is a diagrammatic illustration depicting a representation of an embodiment of the graphical user interface window within the unified presentation layer displaying algorithms included for clustering of intelligent object data within the intelligent object pool (IOP) 204, as well as a number of other property panes.

FIG. 12 is a diagrammatic illustration depicting an embodiment of an interface iPool Zoom viewer for viewing iPool data relationships, utilizing techniques that include dendrograms and self-organizing maps (SOM).

FIG. 13 is a diagrammatic illustration depicting an iPool Zoom interface and display screen image for viewing iPool data relationships, utilizing in this example a Principal Components Analysis (PCA) algorithm.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Having described some of the problems and limitations of conventional systems, architectures, methods, procedures, and computer programs and computer program products in the Background section, we now direct some further attention to highlights of some advantages structural, methodological, and operational features of the invention as compared to such conventional systems and methods. As some comparisons necessarily require or benefit from an understanding of the invention, this comparison is not conveniently provided in the Background section.

Some Problems and Limitations with Conventional Methods

Speed and efficiency for accessing and analysis of global data remains a serious challenge in Life Science and biological computing. Existing software methods have relied on prohibitively expensive computing hardware to provide the heavy lifting required for complex biological computing. Existing software often routes entire raw data files for processing and generally require data locking when shared files are accessed. Mass spectrometry data, for example, may present files of 100 megabytes or more. Additionally, existing techniques for data translation and integration require "wrapping" software that further bloats programming and processing overhead, with resulting performance hits.

To address this data-intensive processing, expensive database back-ends, mainframe computers and / or cluster servers are bundled with functional but inefficient network integration software offered by such companies as Oracle, IBM (DiscoveryLink/DB2) and Compaq (Alpha Systems). These hardware and database back-end resources can range be very costly and even with these massive hardware infrastructures, significant time-delays in result generation remain the norm (Einstein, 2000). Clearly more efficient access, integration and processing techniques, to reduce hardware expense and to speed up results on any system, will be valuable to the industry.

A "functionally integrated" computing environment, wherein meaningfully related data may be directly compared and subjected to unified processing and detailed analysis by potentially heterogeneous applications, remains a goal for Life Science computing. Existing software methods fail to provide for analysis of complex, dynamic analysis utilizing "functionally integrated" analysis of multiple relationships between globally available data, utilizing heterogeneous applications. Existing data structures offer limited attribute definitions, disparate and inflexible annotation standards, and fail to dynamically define attributes for unified

processing in heterogeneous environments. Limitations such as these restrict comparability of heterogeneous data and the dimensionality for functional analysis and viewing.

Companies such as Array Genetics and Nonlinear Dynamics provide tools for detailed analyses of specified structures, activities and their multidimensional inter-relationships within homogenous environments. However, the need to analyze data based on more flexibly defined multidimensional dependencies using heterogeneous data and applications is recognized. For example, Cellomics has recently acquired software which provides the multidimensional analytical functionality to "extract cellular knowledge from multiparameter biological measurements integrated with facets of chemical structure information" (Cellomics, 2001). Another advanced software initiative, by the National Center for Genome Research (NCGR), is currently attempting to meet these goals in a limited manner by developing Integrated SYStems (ISYS) software, that: promotes loose coupling of independent components through a flexible, semistructured data model that supports run-time association of attributes with objects, and allows different components to maintain different "views" of the same object (Seipel et al, 2001).

Like other competitive software, this model fails to provide for data interoperability with diverse applications without manual programming and processing-intensive "wrapping" for translation. Competitive software designed to integrate diverse data generally requires software wrapping to meet translation and interoperability requirements, with resulting processing overhead and dependency on additional human and hardware resources (e.g. DiscoveryLink, Java, CORBA, XML). Even with software wrapping, direct interaction between flexibly associated attributes of diverse data and between this data and diverse applicastions, is limited. Although the goal is recognized, the need remains for software capable of efficiently supporting analysis of complex dynamic relationships, between multiple data dependencies, from diverse global sources.

Detailed security and persistent data auditing, tracking, regulatory validation and alerting remain a problem in global Life Science computing. Existing software methods fail to provide sufficient security and regulatory compliant validation for management and of exchange sensitive biological information over global networks. No company provides byte-level security and access control and few provide the guaranteed raw data integrity required to maintain patient confidentiality and clinical validation over any network.

Leading software technology such as IBM's DB2/DiscoveryLink technology and Oracle's 9i database, for example, fail to provide security to the tightest level. These companies

draw on structured query language (SQL) techniques and provide their finest grained security to the row level.

Virtual Private Database (VPD), an Oracle database feature, offers fine-grained access control or row level security. It allows multiple communities of users to share the same database, yet only access the row(s) of data that pertain to each individual user or community of users. (Oracle Corporation, 2001).

Additionally, only a few companies, such as LabVantage and Telecation provide for network-oriented regulatory validation based on CDER/CFIR/G*P (GLP, cGMP, GCP) compliant data tracking and history. Products that integrate global data resources are unable to maintain detail data state history over networks to maintain compliance with strict laboratory information management (LIMS) protocols. Overall, the current lack of techniques such as tightly defined access controls, assured integrity of raw data, and strict data history and tracking remain a hindrance for data sharing and collaboration, both for new drug development, and especially well for clinical data acquisition and clinical trials success. Improved security techniques are called for in software for Life Sciences.

Some Exemplary Benefits and Advantages

The inventive Sentient IT platform (also called "IMO IT Platform") and architecture, a particular implementation and embodiment of a more generalized and generic IT platform and architecture, and its comprised components and access interfaces provide a number of important benefits to bioinformatics computing in particular, and to distributed and Life Science computing in general. These include benefits for access and processing speed involving terabytes or even petabytes of data, previously unavailable multidimensional analytical functionality, flexible distributed computing, and unmatched methods for security, reliability and regulatory validation.

The Sentient IT Platform as well as other embodiments of the inventive IT platform, offers significant benefits for improved speed and efficiency. This enhanced speed and processing efficiency is made possible by techniques for interactive content routing and direct comparison of data content vectors, as well as meta-data and data annotations, to allow for direct parallel data-to-data communication and self-organizing data. Vector subset "workspace" techniques for access and routing, wherein subsets of data content as small as 1 byte may be specified for queries, minimize network transfer and processing load and provide faster, more meaningful access to terabytes databases.

As is commonly known, increased processing power can increase speed, and the Sentient IT Platform's unique ability to enable a variety of distributed supercomputing infrastructures, such as for grid-based or peer-to-peer (any-to-any) processing will be useful. Biosentients unique methods for efficient data-enabled parallel processing and resource availability monitoring across distributed environments will extend and enhance the speed benefits offered by distributed computing infrastructures.

Additionally, the Sentient IT Platform offers techniques for faster and more efficient data type translation and applications integration and new applications assembly. The current state of the art requires "wrapping" of existing data files for translation and integration, which invariably adds processing overhead. In contrast, the invention provides unique methods for translation based on nested vector look up tables, which generally reduce the processing overhead and improve performance for accessing of existing data files.

In addition to speed benefits, the Sentient IT Platform offers significant benefits for improved analytical functionality across data types. The Sentient IT Platform automates integration of heterogeneous data dependencies and relationships as well as related validated and ranked annotation and literature. Recent techniques for data integration such as by InCellico and Skila provide unified environments for data relationship viewing, clustering and data sharing. In contrast, the Sentient IT Platform enables direct comparison of data content and annotation across data types and structures utilizing heterogeneous analytical applications, for "functionally integrated" analysis of multiple variable interdependencies.

In addition to unmatched functional integration and interconnectivity between heterogeneous related data, the platform provides for easy plug-in of existing analysis tools and automated new applications assembly. Interactive user customization and automated applications assembly include a unified environment for applications component modeling and linking, and efficient automated application integration made possible by a functional module comprised within the Intelligent Object application framework, comprising the Intelligent Object application framework component, automated applications assembly component, application translation interface and object translation engine.

Additional analytical benefits for analysis follow from Sentient IT Platform methods for self-organizing data, distributed data learning, and for applications such as dynamic, kinetic visualization, simulation and prediction based on direct comparison of relevant data content vectors and meta-data to in silico experiments and models. Integrated knowledge aggregation and

distributed learning engines can provide analytical tools for automated, forecasting, alerting and data-driven simulations.

Security benefits include improved control to accessing and routing of data. In contrast to current state of the database art, such as Oracle's "9i"database structure, which provide "finely grained" row-level data access control, the Sentient IT Platform provides unique methods for controlling access to the level of a workspace vector subset as small as a single byte. Even if identification and e-signature encryption techniques enabled by the Sentient IT Platform for unique data, user, session and machine identification and access permission were subverted for an intercepted transmission, the interceptor would normally find only a small subset of data, meaningless outside of the context provided by the Intelligent Object, Intelligent Object Handler and Intelligent Object Pool.

Benefits for reliability include permanently logged, always-on, detailed activity histories for every data object. In contrast to technologies such as Mercury Prime's Razius" and Sparta System's "Trackwise", which required tightly delimited network connectivity scenarios to enable data tracking and auditing, the Sentient IT Platform assures that comprehensive data state activity tracking and history is always available, over flexible network architectures, whether analysis is localized on a single workstation or over wide area or global networks. These improved data management and data state rollback options are made possible by components and interfaces such as the object state engine, status management component, and cache-based non-destructive overlay processing methods. Non-destructive processing methods and one-click integrity assessment assures that raw data integrity is maintained at all times, and any data manipulation can be rolled back to any previous data state.

The Sentient IT Platform also offers unmatched support for regulatory compliance and validation over the complete data and applications life-cycle. The object state engine, status management component and legacy synchronization interface comprise methods for maintaining persistent (always-on or immediately updated) end-to-end data history, automated compliance status reporting and alerting and full (GLP/GCP/GMP/ISO 9000/CDER CFR/CFIR) validation support. Utilization of the application translation interface and Intelligent Object application framework will make it possible to efficiently generate new, or integrate existing interactive clinical and regulatory report interfaces.

Compared to systems for scientific, distributed and biological computing more broadly defined, the Sentient IT Platform provides a number of benefits. For example, the Sentient IT Platform provides unified, real-time bi-directional user interactivity within "any-to-any" network

connectivity environments, an improvement over systems such as internet portals and web-based applications services (ASPs) provided by Entigen's Bionavigator or LabBook's eLabBook. Network options may be automatically configured, according to detected network profiles and custom user preference profiles. Flexible options for network connectivity include distributed grid and peer-to-peer super-computing, along with more traditional client-server, server-server and other custom network connectivity options. These tools enable more powerful collaboration and analytical capabilities as, for the first time, all data and applications may be securely integrated over any network and processing configuration and are instantly available anywhere, any time.

The Sentient IT Platform provides unique methods for data and applications translation and processing, providing an advantage for speed, efficiency and functionality over current systems for data and applications integration such as provided by Lion Biosciences and IBM's DiscoveryLink. These advantages in speed and efficiency parallel the unrealized functional interconnectivity providing marked improvements for direct comparison and analysis of meaningfully related dimensions of data content, even across previously incompatible data types. Vectorized access, unique translation technologies and data-enabled parallel processing provide real time answers, without requiring programming-intensive, inefficient data or database "wrapping", dedicated database technology or massive parallel mainframe / cluster server expenses. For the first time, terabytes of complex, heterogeneous data can be analyzed in an affordable, unified environment, for real-time answers to drug discovery questions.

The Sentient IT Platform provides functional integration of existing analytical and data query tools with new tools and previously incompatible data, without requiring compliance to specific data annotation standards or data warehousing specifications. This is an advantage over traditional analytical tool or data access systems such as SwissProt's Melanie or Edinburgh Biocomputing Systems' MPSRCH. In the Sentient IT Platform, direct vectorized data access and translation look up tables provides highly efficient, validated, normalized data and applications interoperability. For the first time, all existing and new data and applications are integrated for real-time multidimensional analysis of heterogeneous data, in a globally accessible and completely unified user environment.

While the complete Sentient IT Platform offers the widest range of advantages, functional modules comprising one or more processing components and / or access interfaces may be utilized to enable specific functionality in key areas. Together, the Intelligent Object, Intelligent Object Handler and Intelligent Object Pool provide performance, functionality, security and validation previously unavailable to bioinformatics and Life Sciences computing.

Having described some of the features and advantages of the inventive system, method, and information technology (IT) platform, attention is now directed to a more detailed description of a particular embodiment of the inventive IT platform and architecture, referred to as the Intelligent Molecular Object Information Technology Platform (IMO IT), or alternatively in some embodiments as the Sentient Information Technology Platform (Sentient IT).

The inventive Intelligent Molecular Object Information Technology (IMO IT) Platform and its constituent subsystems, methods, procedures, and computer software algorithms, solves the diversified data and format as well as the heterogeneous database and applications problems by integrating unique components, processes, engines and interfaces such as networked data type, database, and application type detection and table generation, coupled with multidimensional table lookup and pointers. These components, processes, engines and interfaces work within the IMO IT architecture at the data object level to provide such advantages as automated, real-time access and translation capabilities, thus enabling instant handling of diverse data in heterogeneous applications environments.

The IMO IT Platform also solves these and other problems and limitations of conventional systems and methods by integrating unique components, processes, engines and interfaces such as workspace vector subset selection, dynamic meta-data indexing at the data object and data subset levels, as well as direct information interchange at the data object and data subset levels. These components, processes, engines and interfaces, work within the IMO IT architecture to provide significant advantages for efficient analysis of high numbers of high density data, allowing for true real-time data acquisition and analysis in a global data environment.

The IMO IT Platform technology, in contrast to conventional existing technologies described in the background, does not reproduce or alter the raw data in any way, thus eliminating data redundancy while simultaneously enabling multiple queries on a single data object simultaneously. By using meta-data reference tables, pointers and tags to provide real-time translation and integration, which efficiently refers only to the aspects of any raw data relevant to a specific query, the IMO IT Platform avoids data redundancy and data access locking requirements. This provides significant advantages over currently existing technologies as described by such companies as Oracle and Sybase, for example, in US 6105030 and US 5832484; each of which is hereby incorporated by reference.

Furthermore, in contrast to the conventional practices requiring retooling and/or translating at application and hardware layers, described in the Background, the IMO IT Platform

enables translation, applications integration and validation of existing systems, in real-time, at the data object level. This allows for efficient scalability, interoperability and applications development without retooling existing systems, and provides for data-enabled validation of existing hardware and software systems.

In one aspect, the inventive methods and procedures remedy these constraints and limitations by providing an architecture allowing interactive, object-based intelligent communication in-between the data itself to extract all relevant content in a fast, unique and automated manner, within complex network environments without the need of upgrading or replacing current computer systems.

These methods and procedures, also remedy these constraints and limitations by allowing interactive, object-based intelligent communication in-between the data itself to extract all relevant content in a fast, unique and automated manner, within complex network environments without the need of upgrading or replacing current computer systems. The **Intelligent Molecular Object 200** technology provides a flexible global standard, which allows for seamless integration and real-time answers to complex, multidimensional and interdependent queries. The intelligent molecular object technology provides a framework for scale-up and dynamically changing application needs in bioinformatics and the life sciences.

They also remedy these constraints by allowing interactive, object-based intelligent communication between object data based on an unified presentation layer, an user definition and administration shell and an automated application/database definition generator interface which accounts for seamless integration into the intelligent molecular object technology.

They further remedy these constraints by allowing interactive, parallel, object-based intelligent communication in-between the data itself to extract all relevant content in a fast, unique and automated manner, within complex network environments without the need of upgrading or replacing current computer systems. The intelligent molecular object pool technology provides seamless integration and real-time answers to complex, multidimensional and interdependent queries, but still maintains boundaries of data subsets to govern a secure, inter-data communicative global access. The intelligent molecular object pool technology also provides a framework for scale-up and dynamically changing application needs in bioinformatics and the life sciences.

Having now described some of the features, benefits, and advantages of the invention, attention is now directed to descriptions of various facets and embodiments of the invention.

Some of the major subsystems and procedures contributing to this operation and the advantages that follow therefrom are now described in greater detail.

Software Architecture For An Information Technology Platform Using Intelligent Objects

The software architecture for an information technology platform using intelligent objects such as Intelligent Molecular Object (IMO), software and processing method addresses all or selected steps ad procedures of data processing from data acquisition through diverse sources and instrumentation to final output of diverse data analysis results.

In the first step, this technology uses engine components to standardize and/or normalize data from diverse sources to make those otherwise different data comparable. The engines provide a means to accurately compare, thus allowing one to test for identity, increase or decrease in values or functional relevancy and the like. In particular, algorithms for tracking and normalization of object data (Object Normalization Engine, ONE) or image data (Global Image Normalization, GIN) are provided. These algorithms allow one to extract variable and non-variable regions within a set of data and generate a global standard to which all data can be referred. By applying these algorithms, adjustments to all necessary parameters in a multidimensional data set can be made automatically and simultaneously. Within the described architecture, these algorithms can be applied in external modules or plug-in-format for other applications as well as for access via intranet or Internet.

Next, the architecture uses uniquely defined data objects, the "**Intelligent Molecular Object 200**" (IMO) as its data records. Each the object consists of sets of functional layers (property panes). These property panes are activated or disabled via a Property Pane Controller (PPC). Each object is represented by a Unique Identifiers (UID), which governs object data security and access permissions via the Object Access Manager (OAM), and additional layers which define origin of the object within the network (Object Root Router, ORR) and route content and results interactively (Interactive Content Router, ICR) across the network in a standardized fashion. A Status Management Component (SMC) monitors data integrity, command history and GLP/GMP-compliance via a table-based Object State Engine (OSE). The object state engine provides control for any object activity in real-time, logs activity records, provides GLP/GMP compliant experiment state assignment, state management for object data on the network, information request ranking and vectorized, direct addressing of data subsets (Vector Subsets, VSS) to minimize network traffic. This processing engine also handles external query submission and result synchronization for inter-object queries. All these processes allow the

object to communicate intelligently via VSS, and keep track of their interactions. Using Meta-data Indices (MDX), workspace-oriented VSS and Object Pane Descriptors (OPD) accounts for quick and direct communication with diversified applications and databases via an Object Query Interface (OQI). An integrated Application Translator Link (ATL) communicating via the OAM allows for application integration. Additionally, functionality is provided for Direct Information Interchange (DII) between objects, graphical preview of the object data (Object Graph Preview, OGP) and raw data matrix structure description (Raw Data Matrix, RDM; Matrix Structure Definition, MSD).

Next, the architecture provides an applications framework within the IMO technology, the **Intelligent Object Handler 202** (IOH), describing sets of user interactions and object environment definition protocols. This **IOH 202** is comprised in general of a set of processing engines and access interface protocols. These protocols provide methods and functions for preparation and presentation of data objects (IMO Application Framework, IMO-A) for interaction within heterogeneous environments. A Unified Presentation Layer (UPL) within the **IOH 202** provides a web-enabled Graphical User Interface (GUI) to integrate components or modules from diverse applications, laboratory systems environments, and to act as handler for IMO data (IMO Handler, IMO-H). Additional components include a User Definition Administration Shell (UDA), a Master Query Component (MQC) and an interface to automate the query of application and database requirements via an Application Definition Generator (ADG). The creation and initiation of new IMO data is provided via an integrated IMO Generator (IMO-G). A Direct Instrument Acquisition & Control Interface (DIAC) and a component for Automated Application Assembly (AAA) provide integration of real-time data acquisition and analysis. Data Type Translators (DTT) are provided to integrate automated transformation from heterogeneous data sources into IMO data in real-time. Automated normalization of data by calibration with standardized empirical criteria within the workspace IMO Standardization Technique (IMO-S) is managed through integrated meta-data tags and pointers. Several access interfaces are also integrated at the IOH level. Next, an Object Translation Engine (OTE) is integrated, which governs interactions (such as transformation, integration and information access) between IMO data and other diverse data environments to enable real-time communication. Such processes automatically determine other data structures, look up functional information of the data, create descriptors which correspond to object property panes and determine application type and access to IMO's in real-time. All the engines and interfaces establish the connection to the legacy world and provide bi-directional, multidimensional, secure

access to applications (Application Translation Interface, ATI), for queries via OQI, for result generation (Result Generation Interface, RGI) and for synchronization with offline legacy systems (Legacy Synchronization Interface, LSI). Additionally, learning engines such as a Knowledge Extraction Engine (KEE) or Distributed Learning Engine (DLE) and the like can be implemented within the IOH **202**.

Next, the architecture provides processes, which govern a global virtual data pool, the Intelligent Object Pool **(IOP) 204**. The processes contain definitions for subsets, called Intra-Pools (iPools) **155** regulated by boundary protocols, which define integrity and persistence of IMO relationships. IOP components comprise iPool boundary interfaces, iPool meta-data query and content access interfaces and iPool content ordering definitions and protocols. iPool boundary interfaces include the iPool Security Authentication (iPSA) component, which provides security authentication; the iPool Integrity Assessment (iPIA) and Object Integrity Assessment (OIA) components, which provide data integrity and persistence; and the iPool Availability Monitoring (iPAM) and iPool Exchange Protocols (iPEP) components, which define and control availability and exchange at the IMO level. IPool Meta-data query definition interfaces are provided by Object-to-Object Query Meta-data (OQM), Real-Time Meta-data Link (RML), iPool Meta-data Index (iMDX), and iPool-to-iPool Query (PPQ) components. These processes apply interactive presorting and exclusion algorithms, provide object clustering, object result clustering and object-to-object interaction rules, and enable rapid, relevant data access via real-time meta-data queries ordered within the iPool on the IMO level. iPool content access and ordering definitions and protocols include an Aggregate Meta-data Index (aMDX), Aggregate Real-time Significance Generator (aRSG) which integrate result merging algorithms and real-time answer generation. These iPool content access protocols and order definitions allow for object-to-analysis tools interactions, real-time result aggregation and real-time exclusion of irrelevant object data layers. Additional components include an IMO Zoomer (IMO-Z), which defines the proximity of individual IMO data within the IOP and enables multidimensional IMO data viewing and functional ranking.

Through the functionality detailed above, the IOP provides an engine for global result aggregation (Result Aggregation Engine, RAE) and instant answer output across diversified data subsets and an interface to assess integrity of **iPools 155** within the virtual, global data pool.

Engines, interfaces and components comprising methods, functions, and definitions are provided, to define and describe a unique, data-enabling software architecture (IMO IT Platform). These engines, interfaces and components implement an information technology platform which

utilizes Intelligent Molecular Object (IMO™) data and consists of a common framework comprising always-on as well as event-driven processing engines, access interfaces, plug-in modules and other components.

The IMO IT Platform architecture defined and described below addresses all steps of data processing from data acquisition through diverse sources and instrumentation to final output of diverse data analysis results reports.

The IMO IT Platform utilizes uniquely defined data objects, "Intelligent Molecular Object" data objects as its data records. Each IMO consists of sets of functional layers (property panes), describing content and providing certain functionalities to the object. These property panes are dynamically activated or disabled via a Property Pane Controller (PPC), function of which is to allow or block access based on user privileges, data pool definitions and the like.

Each IMO is represented by an Unique Identifier (UID) contained within the identity pane, so it can be addressed and identified on any network directly via its ID. The identity pane also governs object data security and access permissions via the Object Access Manager (OAM), an integrated part of the PPC to initiate object communication. Next, each IMO contains a layer, which contains information defining the origin of the object within the network (Object Root Router, ORR) and its owner. Next, each IMO contains a layer, which routes content and results interactively (Interactive Content Router, ICR) across the network using standardized protocols.

Next, a Status Management Component (SMC) monitors data integrity and command history in GLP/GMP-compliance via a table-based Object State Engine (OSE). The Object State Engine consists of processes which control any object activity in real-time, log activity records, provide GLP/GMP compliant experiment state assignment and state management for object data on stateless networks. In addition, the Object State Engine handles information request ranking and vectorized, direct addressing of data subsets (Vector Subsets, VSS) to minimize network traffic. This processing engine also handles external query submission and result synchronization for inter-object queries by providing routing, property pane access clearance and direct, workspace-oriented VSS addressing. All these processes allow the object to communicate intelligently via VSS, and keep track of their interactions.

Next, the IMO includes Meta-data Indices (MDX) layer for rapid access, and Object Pane Descriptors (OPD) which allow for quick and direct communication with diversified applications and databases via an Object Query Interface (OQI), which allows for object-level direct Information Interchange (DII) between objects.

An integrated Application Translator Link (ATL), communicating via the OAM and ICR, accounts for integration of external applications and/or remote application status inquiries. Next, functionality is provided for graphical preview of the object data (Object Graph Preview, OGP) and raw data matrix structure description (Raw Data Matrix, RDM; Matrix Structure Definition, MSD).

The architecture defines an applications framework within the IMO technology, the **Intelligent Object Handler 202** (IOH), which provides sets of user interactions and object environment definition protocols for the IMO data. This **IOH 202** is comprised in general of a set of processing engines and access interfaces. A Unified Presentation Layer (UPL) within the **IOH 202** provides a web-enabled Graphical User Interface (GUI), which integrates data, components and/or analytical and processing modules from diverse applications and laboratory systems environments. In general, these protocols provide for preparation and presentation of data objects for interaction within heterogeneous environments (IMO Application Framework, IMO-A).

To ensure automated, real-time normalization of data using one or several calibrations with empirical criteria within the workspace, the IMO Standardization Technique (IMO-S) is provided, which activates engine components for standardization and normalization through utilization of integrated meta-data tags and VSS pointers. These engine components are defined by the following methods and functions, which standardize and/or normalize data from diverse sources to make those otherwise different data comparable. The engines provide a means to accurately compare, thus allowing to test for identity, increase or decrease in values or functional relevancy and the like. The following algorithms for tracking, standardization and/or normalization of object data (Object Normalization Engine, ONE) or image data (Global Image Normalization, GIN) are defined. These algorithms allow for the extraction of variable and non-variable regions within a set of data and generate a global standard to which all data can be referred.

In the case of ONE, which is described in but not restricted to the following example in the field of Life Sciences, the data are comprised of numeric matrices, text annotations, chemical structure information, chirality information, spectral information, sequence information and the like.

In the case of GIN, described by but not limited to the following example in the field of Life Sciences, the data are information contained in fluorescent and/or otherwise visibly stained 1D or 2D gel electrophoresis images, array images, microscopic images and the like, all of which may differ in image acquisition parameters, detection technique, intensity, color, positional

distortions of zones, bands, spot or other regular or irregular objects contained in the images which, in consequence, relate to certain macromolecule properties, such as size/molecular weight, isoelectric point, concentration, biological activity and the like. By applying algorithms, which define a vector subset for workspace selection in a single or a set of images, only those, typically relatively small areas of the images are processed which are needed to achieve relevant comparison. Such, data transfer between different objects and for temporary storage and processing via normalization, is reduced to the subsets for speed and efficiency. By reference to either a common, global standard or a dynamically obtained, averaged reference from all data included within the comparison query, adjustments to all necessary parameters in a multidimensional data set can be made automatically and simultaneously in parallel. Since those processes apply only to temporarily extracted small data subsets, several different request on the same object may be processed at the same time. Since in each case only vector subset are generated, no raw data alteration occurs and GLP/GMP-compliant data integrity is maintained. Within the described architecture, the algorithms can be applied as processing engines, in external modules or in plug-in-formats for other applications as well as for remote access via intranet or Internet.

Engines, interfaces and components provided within the **IOH 202** to provide an integrated analytical framework, including but not limited to the following. A User Definition Administration Shell (UDA) interface is provided, which creates, modifies and administers user profiles and privileges and defines rules for users within individual subsets of data, called Intra-Pools (iPools) **155**, as well as group memberships and topic-related access rights. A Master Query Component (MQC), is provided, which creates complex, multidimensional queries, containing pre-defined, configurable subsets of forms commonly used, but not restricted to, in diverse areas of Life Sciences. An Application Definition Generator (ADG) component is provided, which automates the query of application and database requirements and is comprised within related translation components to generate tables required for integrated real-time property pane presentation at the data object level. An IMO Generator (IMO-G) component is provided, which creates new IMO data from existing data resources, or from newly acquired instrumentation data. An IMO handler (IMO-H) component is provided, which initiates user commands and queries at the IMO level via the use of integrated meta-data tags and pointers. A Direct Instrument Acquisition & Control Interface (DIAC) is provided, which enables bi-directional real-time communication between the **IOH 202**, the IMO **200** and diverse instrumentation. An Automated Application Assembly (AAA) component is provided, which enables integration of real-time data

acquisition and analysis functionality through just-in-time (JIT) module linking. A Data Type Translation (DTT) component is provided, which integrates translation tables from the ADG from heterogeneous data sources into IMO data in real-time. The DTT are comprised of dynamically generated sets of reference tables to provide rapid access through data structure definitions.

Several engines and access interfaces integrated at the IOH level are defined, which utilize the meta-data tags and pointers to pass information between internal and external components. All the engines and interfaces establish the connection to the legacy world. An Object Translation Engine (OTE) is included, which governs interactions (such as transformation, integration and information access) between intelligent object data and other diverse data environments to enable real-time communication. Such processes automatically determine other data structures, look up functional information of the data, create descriptors which correspond to object property panes and determine application type and access to IMO's in real-time. An Application Translation Interface, (ATI) is defined, which provides bi-directional, multidimensional, secure access to applications for queries via OQI. A Result Generation Interface (RGI) is defined, which provides validated, assembled, ranked and tabulated results to the RAE, enabling the generation of output reports across diversified data subsets. A Legacy Synchronization Interface (LSI) is defined, which provides synchronization with offline legacy data. Additionally, learning engines such as a Knowledge Extraction Engine (KEE) or Distributed Learning Engine (DLE) and the like can be implemented within the **IOH 202**.

Next, the architecture provides the following interfaces, component processes and engines which enable and govern a global virtual data pool comprised of IMO data, the Intelligent Object Pool **(IOP) 204**. The component processes contain definitions for subsets, called Intra-Pools (iPools) **155**, regulated by boundary protocols, which define integrity and persistence of IMO relationships. IOP components are defined, which comprise iPool boundary interfaces, iPool meta-data query and content access interfaces and iPool content ordering definitions and protocols.

iPool boundary interfaces are defined, which include the iPool Security Authentication (iPSA) component, which provides security authentication; the iPool Integrity Assessment (iPIA) and Object Integrity Assessment (OIA) components, which provide data integrity and persistence; and the iPool Availability Monitoring (iPAM) and iPool Exchange Protocols (iPEP) components, which define and control availability and exchange at the IMO level.

iPool Meta-data query definition interfaces are provided by Object-to-Object Query Meta-data (OQM), Real-Time Meta-data Link (RML), iPool Meta-data Index (iMDX), and iPool-

to-iPool Query (PPQ) components. These processes apply interactive presorting and exclusion algorithms, provide object clustering, object result clustering and object-to-object interaction rules, and enable rapid, relevant data access via real-time meta-data queries ordered within the iPool on the IMO level.

iPool content access and ordering definitions and protocols are provided, which include an Aggregate Meta-data Index (aMDX), Aggregate Real-time Significance Generator (aRSG) which integrate result merging algorithms and real-time answer generation. These iPool content access protocols and order definitions allow for object-to-analysis tools interactions, real-time result aggregation and real-time exclusion of irrelevant object data layers. Additional components include an IMO Zoomer (IMO-Z), which defines the proximity of individual IMO data within the IOP and enables multidimensional IMO data viewing and functional ranking.

It is evident from the above description, that this IMO IT architecture allows for decision empowering, real-time answers to complex, multidimensional, interdependent queries by providing the infrastructure for a global, comprehensive analysis of otherwise not accessible vast, inconsistent sets of data.

The following examples are offered by way of illustration and not by way of limitation. In a typical life sciences example, a 2-dimensional electrophoretic protein separation is carried out and the silver-stained separation pattern is introduced in an imaging workstation to obtain a high resolution, high dynamic range image representation of the spot pattern (typically, about 2500 – 6000 individual protein spots/image). Each of those spots represents a single protein at a defined expression stage in a specific cell environment, e.g. in this example, human liver cells. These pattern reflect also genetic differentiations and/or modifications, e.g. in case of human samples, the origin of the cell, gender, age, race, physical condition of the individual and the like.

It is obvious from the above, that such pattern represent in inherent multidimensional complexity, all of which even more expressed by laboratory-to-laboratory deviations in performing the analytical procedure, sometimes also on different types of instrumentation. To analyze such a pattern for, for instance the increase or decrease in concentration based on a pathological disease condition or for drug-induced changes on certain proteins in enzyme-, immunological activity and the like, non-patient specific data have to be separated from those common in all pattern. A large series of such gels must be analyzed, standardized and compared to achieve this goal.

In the example, the IMO platform technology will do the following: A scientist on a laptop (1) in site A in the USA logs onto the IMO platform; the UDA within the **IOH 202** verifies

the login, sets user permissions and encryption level, generates a session ID and starts a session for (1) in the **IOH 202**. Next, the scientist (1) uses the menu in the UPL to acquire instrument data, in this example gel image data. The DIAC within the **IOH 202** communicates with the imaging workstation (2) to remotely start image acquisition, the IMO-G creates a new data objects for the currently acquired image. This new IMO consists now of a UID pane, which uniquely identifies the IMO via a 10-character key across the network. The UID pane also defines object type, contains information about the origin (ORR) on the network (on 1), the owner (creator) of the IMO, how content (the raw image data in this case) is routed (ICR) interactively on the network (in this case, the raw image file at the imaging workstation (on 2)), of an OSE pane, in which the first 2 entries are created via the SMC (an object creation record, and the current data state entry, "data acquisition in progress", an indexed multi-digit number from the GLP/GMP state assessment table), a OPD pane describing which panes are functionally defined, a dynamically updated OGP (displaying a progressive thumbnail view of the image in real-time as it is acquired) and a RDM pane consisting essentially only of file type description and pointers to the original image file, thus there is no data redundancy anywhere on the entire network. When image acquisition is complete, the SMC on the IMO adds another status entry "data acquisition completed" to the IMO state history log and adds final data MSD's to the RDM pane.

Simultaneously, another scientist (3) in site 2 in Japan logs onto the IMO platform; the UDA within the **IOH 202** verifies the login, sets user permissions and encryption level, generates a session ID and starts a session for (3) in the **IOH 202**. Next, the scientist (3) uses the menu in the UPL to submit a global query about gel image data, in this case, describing a defined disease-related change in protein expression of peptides with an iso-electric points around 5.5 and a molecular size ranging between 80k DA – 120k DA in liver cells from white, human males with an age above 50. The MQC within the **IOH 202** analyzes the query by providing the IMO-H with VSS for the appropriate workspace definitions within the 2D gel image data (based on the entered pI and size constraints, only a small fraction of each images is relevant to the query), which, in turn, gates the query to IMO's via OQI. Based on MDX links, the PPC will only inquire those **iPools 155** and IMO's with matching OPD's. The OAM on each such IMO checks access privileges, triggers the PPC, sends update request to the OSE, extracts partial image information from the original RDM for temporary storage to be processed via ONE and GIN for image workspace normalization and comparison. Standardized comparisons (in this case, the protein spot location in the individual IMO RDM subset, and its concentration, represented by its integrated, optical density calibrated intensity) within the VSS are reported back via ICR in each

IMO to the RGI at the IOH level, which creates ranked output. Similarly, MDX-based, the OTE at the IOH level performs data translation of workspace data from other databases using the ATI to integrate those relevant data into the result-ranking. During this time, another scientist (4) in site B in the USA logs onto to the IMO platform, and performs a similar, but not identical query. The DLE immediately addresses the MQC to define a new VSS only within the already addressed, active IMO's RDX's. The RAE at the IOP aggregates the IMO results, sorts them based on relevancy and similarity, reports to the KEE and DLE for dynamic MDX update, and posts the final tabulated answer back to the **IOH 202** via RGI. Proximity of integrated data is related to the IMO-Z for graphical representation. The UPL now presents the aggregated, ranked answers to the query in several graphical and numerical windows to the scientist (3), including the newly generated image from scientist (1). On a different location and network, at the same time the local UPL in scientist's (4) computer displays the aggregated answer for his query. In the above example, for scientist (1) 212 peptides and for scientist (4) 96 peptides were identified, which differ in their expression level, amongst a set of > 500000 relevant 2D gel images with approx. 3000 individual spots each.

It is noted, that in a conventional database environment, processing of the query would require the analysis of the entire gel images and their annotations, at the required resolution typically at file sizes around 2-5 MB each, and the alignment (normalization) of the image in its entity, thus, requiring enormous amounts of data to be processed. The IMO IT platform technology's dynamic workspace definitions via VSS reduces this so significantly, that alignment, matching and comparison are reduced to, in this example, about 40-60kB data each (or approximately, by 60-fold), allowing real-time result aggregation.

In light of the description provided here and in the following more detailed description as well as the appurtenant figures, it will be appreciated that the invention provides an architecture, business model and method of doing business related to searching for and analyzing data generally and in particular relative to biological, chemical, and life sciences type information. Several exemplary embodiments are now described by way of example, but not limitation.

In one aspect, the invention provides a software architecture for an information technology platform, comprising of always-on and event-driven, engines, interfaces and processes and using intelligent molecular software data objects for interactive data records.

In another embodiment, this architecture is further defined such that the architecture further comprises one or more of: a. an **Intelligent Molecular Object 200** (IMO), a versatile, data-enabling software object, which provides for real-time translation, integration, and object-to-

object/object-to-analysis tools communication at the object data level, to allow multidimensional, platform-independent complex queries in real-time; b. an **Intelligent Object Handler 202** (IOH), which provides the application framework and user interface for IMO data, to allow for seamless integration of their benefits into legacy systems; and c. an Intelligent Object Pool **(IOP) 204**, comprising one global virtual data pool comprised of IMO data, which integrates diverse data resources on any system or network to provide result aggregation and instant answers across diversified data subsets.

In another aspect, the IMP of the architectural method and platform is further defined to include one or more of: a. a unique identifier (UID), comprising a property pane layer created at IMO generation, which provides typically a 10 byte, alphanumeric unique identification on any system or network; b. an object access manager (OAM), a property pane layer which governs data security and access according to UID permissions; c. an object root router (ORR), a property pane layer which contains information to define the origin of the object within the system or network; d. an interactive content router (ICR), a property pane layer which routes content and results interactively across the system or network; e. a status management component (SMC), comprised of an object state engine and certain interfaces, which monitors data integrity and command history in GLP/GMP-compliance via state history and governs table lookup actions via the ICR; f. a property pane controller (PPC), which controls the initiation of IMO communication according to activation by 3a through 3d, above; g. vector subsets (VSS) for automatic, dynamic, or user-defined workspace definitions, which provide vectorized, direct addressing of data subsets for the ICR to minimize network traffic; h. meta-data indices (MDX), to provide efficient access via dynamically updated meta-data description relevant to extant data queries and definitions; i. object pane descriptors (OPD), which provide information about each object property pane and their function as required for direct communication with diversified applications and databases; j. an interface for direct information interchange (DII), which provides the interface to communication at the object level; k. an application translator link (ATL), which activates the OAM and ICR to determine the property panes for functional presentation and access within a given application or database environment; l. an object graph preview (OGP) pane, comprising a limited resolution image and graphics viewer for quick graphical data review, particularly of image data and spectral datasets; m. a raw data matrix (RDM), comprising a property pane which provides the full information subset for any data format or structure; and, n. matrix structure definitions (MSD), which allows for data field mapping and enables vector access to specific data fields.

In another aspect, the architecture is further defined such that the architecture and accompanying process and method include the **IOH 202** where the IOH further includes: a. a unified presentation layer (UPL), which provides a web-enabled graphical user interface (GUI) to integrate components and/or modules from diverse applications, laboratory systems environments and to act as handler for IMO data; b. a user definition administration shell (UDA), which sets ups and governs access privileges to individual IMO data at the user-defined level and is accessible within heterogeneous network environments; c. at least one engine for data object normalization and standardization, image normalization and standardization, IMO data translation, distributed learning, and knowledge extraction; d. at least one access interface to and in-between instruments, data and applications, comprising interfaces which include, but are not limited to, direct instrument acquisition and control, application translation, direct object query, result generation, and legacy synchronization; e. a master query component (MQC), create complex, multidimensional queries, containing pre-defined, configurable subsets of forms commonly used, but not restricted to, in diverse areas of Life Sciences; f. an IMO generator (IMO-G), an event-driven component to acquire data from heterogeneous data resources, including from ongoing data acquisition, in real-time and transforms device outputs and heterogeneous data types to IMO data; g. an IMO handler (IMO-H), which enables user management of IMO data utilizing integrated meta-data tags and pointers; h. an IMO application framework (IMO-A), which provides integration and access protocols to heterogeneous applications and databases on the object level; i. an application definition generator (ADG), which automates the query and generation of application and defines computing environments for the IMO data translation; j. at least one data type translator (DTT), which define the data type dependencies for the IMO-G according to the applications and database environments defined by the ADG; and k. an automated application assembly component (AAA), which provides for just-in-time (JIT) module linking.

In yet another embodiment or aspect, the architecture is further defined such that the IOP includes one or more of: a. sets of Intra-Pools (iPools) **155**, regulated by boundary protocols, which provide data subset management and the define integrity and persistence of IMO relationships; b. iPool security authentication protocols (iPSA), which authenticates iPool data requests according to user login and object data identification; c. iPool availability monitoring protocols (iPAM), which define the iPool availability and access requirements of diverse data subsets; d. iPool exchange protocols (iPEP), which determine and govern iPool data exchange protocols according to user-defined criteria; e. an object integrity assessment component (OIA),

which assess object integrity for security and QA/QC; f. sets of engines and interfaces to access and generate ranked results within the IOP, including but not limited to an integrity assessment interface, a real-time meta-data linking interface and an iPool-to-iPool query interface; g. an iPool meta-data index (iMDX), which provides dynamic, automated, and user-defined meta-data indices at the iPool level; h. an aggregate meta-data index (aMDX), which provides dynamic, automated, and user-defined meta-data indices at the aggregate IMO level, inclusive of all relevant data resources; i. an object-to-object query meta-data sorter (OQM), to generate temporary tables based on dynamic, automated, and user-defined meta-data indices; and j. an aggregate real-time significance generator (aRSG), which provides for significance detection of values based on query parameters, meta-data indices when relevant, and IMO data ranking.

In one additional particular embodiment, the engines within the architectural platform and method are further defined to include: a. an object state engine (OSE), which provides a continuously-running (always on) set of processes, which monitor and govern activities of IMO data, performing real-time recording, updating and logging functions in GLP/GMP-compliant format.

In another alternative embodiment, one or more engines within the architecture are further defined to include one or more of: a. a set of IMO standardization techniques (IMO-S), comprising engines which provide algorithms for tracking, standardization and/or normalization of object data; b. an generic object normalization engine (ONE), which extracts variable and non-variable regions within any set of object data and generates a global standard to which all data can be referred; c. an engine for global image normalization (GIN), which extracts variable and non-variable regions within any set of image data and generates a global standard to which all data can be referred; d. an object translation engine (OTE), which is comprised of methods and functions for real-time meta-data extraction and table generation of raw data matrix, data object, data field, data structure, data functional information, data type, database type, and application type definitions for the OPD; e. a distributed learning engine (DLE), which provides algorithms for dynamic, automated, and user-defined hypothesis generation utilizing global data resources; and, f. a knowledge extraction engine (KEE), which provides algorithms for dynamic, automated, and user-defined significance discovery and report generation.

In a further embodiment of the architecture, the engines may include: a. a result aggregation engine **(RAE) 224, 475**, to validate, assemble, rank and tabulate results passed from the **IOH 202** and to generate output reports across diversified data subsets.

In yet a further embodiment of the architecture, the interfaces may include: a. a direct information interchange interface (DII), which allows for rapid analysis and results aggregation by providing the interface for object-to-object and object-to-analysis tools via such related interfaces and engines including, but not limited to the OQI, OTE, and the DLE.

In yet a further embodiment of the architecture, the interfaces may include one or more of: a. a graphical user interface (GUI), utilizing web-enabling standards including but not limited to Java and XML; b. a direct instrument acquisition and control interface (DIAC), which provides bi-directional real-time communication between the **IOH 202**, the IMO **200** and diverse instrumentation; c. an application translation interface (ATI), to provide automated real-time detection of diverse data and applications and gate bi-directional access to the OTE, thus enabling functional, standardized integration of IMO data within heterogeneous data and applications environments; d. an object query interface (OQI), comprising an interface for direct information interchange (DII) with IMO data, which initiates query analysis and results aggregation; e. a result generation interface (RGI) to provide validated, assembled, ranked and tabulated results to the RAE, thus enabling the generation of output reports across diversified data subsets; and f. a legacy synchronization interface (LSI), to provide persistence and synchronization of offline legacy data.

In still another embodiment of the architecture, the interfaces are further defined to include one or more of: a. an iPool integrity assessment interface (iPIA), to asses data integrity within a defined iPool for security and QA/QC; b. a real-time meta-data link interface (RML), which provides for rapid relevant data access based on query parameters and MDX information; c. a pool-to-pool query interface (PPQ), which provides for query optimization based on query parameters and relevant iPool data and meta-data intercommunication; and, d. an IMO Zoomer (IMO-Z), which defines proximity and functional ranking of individual IMO data within the IOP and enables multidimensional IMO data viewing to represent object relationships within the pool and in relationship to other **iPools 155**.

In another embodiment, the invention provides a software or combination software and hardware architectural platform that uses objects for real-time, efficient, multidimensional, interdependent intelligent queries.

In yet another embodiment, this software architectural platform and the associated methods and procedures are implemented on general purpose computers, information appliances, and the like information and/or computation devices, at least some of which are coupled to similar devices and servers on an interconnected network, such as the Internet.

When implemented as software, the software may be resident within a memory of the computer or information device and execute within a processor, microprocessor, or CPU of such device. Any conventional computer or information appliance having suitable memory, processor, and interface capabilities may be used, many of which types are known in the art.

An alphabetic list of IMO Information Technology Platform acronyms as used in this description is listed below. In some instances these acronyms are modified by prefixes or suffixes but the meaning is generally clear from the description or drawing in which it appears and all variations are not provided in this in the interest of brevity.

AAA	**Automated Application Assembly**
ADG	**Application Definition Generator**
aMDX	**Aggregate Meta-data Index**
aRSG	**Aggregate Real-time Significance Generator**
ATI	**Application Translation Interface**
ATL	**Application Translator Link**
DIAC	**Direct Instrument Acquisition & Control Interface**
DII	**Direct Information Interchange**
DLE	**Distributed Learning Engine**
DTT	**Data Type Translators**
GIN	**Global Image Normalization**
GUI	**Graphical User Interface**
ICR	**Interactive Content Router**
iMDX	**iPool Metadata Index**
IMO	**Intelligent Molecular Object**
IMO-A	**IMO Application Framework**
IMO-G	**IMO Generator**
IMO-H	**IMO Handler**
IMO-S	**IMO Standardization Technique**
IMO-Z	**IMO Zoomer**
IOH	**Intelligent Object Handler**
IOP	**Intelligent Object Pool**
iPAM	**iPool Availability Monitoring**
iPEP	**iPool Exchange Protocols**

iPIA	**iPool Integrity Assessment**
iPool	**Intra-Pool (data subsets)**
iPSA	**iPool Security Authentication**
KEE	**Knowledge Extraction Engine**
LSI	**Legacy Synchronization Interface**
MDX	**Meta-data Indices**
MQC	**Master Query Component**
MSD	**Matrix Structure Definition**
OAM	**Object Access Manager**
OGP	**Object Graph Preview**
OIA	**Object Integrity Assessment**
ONE	**Object Normalization Engine**
OPD	**Object Pane Descriptors**
OQI	**Object Query Interface**
OQM	**Object-to-Object Query Meta-data**
ORR	**Object Root Router**
OSE	**Object State Engine**
OTE	**Object Translation Engine**
PPC	**Property Pane Controller**
PPQ	**Pool-to-Pool Query**
RAE	**Result Aggregation Engine**
RDM	**Raw Data Matrix**
RGI	**Result Generation Interface**
RML	**Real-Time Metadata Link**
SMC	**Status Management Component**
UDA	**User Definition and Administration Shell**
UID	**Unique Identifier**
UPL	**Unified Presentation Layer**
VSS	**Vector Subsets**

Attention is now directed to particular embodiments of the inventive structure and method, including exemplary embodiments of the intelligent object pool (IOP) as well as aspects and embodiments of intelligent objects (IMO) and intelligent object handler (IOH).

System, methods, procedures, computer programs and computer program products are provided that define and describe exemplary embodiments of an information technology platform system and architecture utilizing an Intelligent Object or Intelligent Molecular Object **(IMO) 200**, Intelligent Object handler **(IOH) 202** and Intelligent Object Pool **(IOP) 204** and Intelligent Object data structures as core processing components.

At least in part because textual description is a linear or sequential process, the inventive methods, procedures, and techniques as well as computer program and operation where such methods, procedures, and/or techniques are implemented in such computer programs, these methods, components and processes will be described in a fashion that may not always describe every facet of an action, operation, or process as it occurs when such are simultaneous or concurrent, and/or interactive actions as they occur. However, it should be noted that the system and method herein described includes bi-directionally or multi-directionally interactive components, processes, and interfaces which perform certain tasks simultaneously, concurrently with overlap, or in a rapidly alternating fashion.

Examples of exemplary enabling computer software or firmware program code are provided to define and describe a particular exemplary embodiment, which utilizes Microsoft C++ as the exemplary programming language. Additionally, software development tools for some embodiments include Visual C++, Microsoft Foundation Classes (MFC), DIB image transformations and matrix-based graphical content generation. The features, capabilities, methods, and structures described herein are not limited to any particular programming language or code, and those workers having ordinary skill in the art will in conjunction with the description provided herein appreciate that other conventional programming languages may be used, and that as new programming languages appear, such languages will also likely support the inventive methods and procedures described herein. It will also be appreciated that the overall architecture, its application across varied domains, its processing engines and its access interfaces are in no way limited to the utilization of Microsoft C++ or the Windows 32-bit operating system environment. It will further be appreciated that other enabling software codes or enabling techniques may also be used, including for example Java, XML and other markup languages, and/or other similar techniques; and that other operating systems such as UNIX-based, Linux-based, or Windows-based, Macintosh Apple-based, or other operating systems may be utilized and/or are interoperable with aspects and features of the inventive system and method.

FIG. 1A is a diagrammatic representation of an embodiment of a general IMO IT architecture, depicting relationships between framework, component engines, interfaces and external data and applications resources.

FIG. 1B is a diagrammatic representation of another embodiment of the IMO IT architecture having different localizations of components and interfaces, and focusing on the integrated data and applications handling aspect of the IMO IT Platform.

FIG. 2 is a diagrammatic representation of an embodiment of the invention showing some functional relationships between instruments, applications, and intelligent object pool (IOP) 204 and diverse databases within an embodiment of the intelligent molecular object (IMO) architecture and topology.

FIG. 3 is a diagrammatic representation of an embodiment of a typical multi-user network collaborative research effort involving geographically diverse data acquisition and query sites.

FIG. 4 is a diagrammatic representation of a flow chart of typical data processing within the IMO IT architecture.

The illustration in FIG. 5 is a diagrammatic representation of an exemplary hardware configuration for the intelligent object pool (IOP) 204 and its enabling architecture. All major elements within the diagram below may be bi-directionally connected over a variety of network protocols. The minimum hardware requirement is defined by a single machine. In an exemplary embodiment, as below, two laptop computers are connected in a client/server configuration to a workstation, to each other in a peer-to-peer manner, and via the workstation directly to a laboratory instrument, such as a gene sequencer or gel electrophoresis machine.

With further reference to FIG. 5, dotted bi-directional lines 248 represent options for "any-to-any" network connectivity enabled via use of intelligent objects as central accessing and routing components. Any-to-any options include but are not limited to LAN, WAN, peer-to-peer (e.g. data, applications, memory and processor sharing between two or more laptops, workstations, etc.), server-server, Portal, ASP and other unified, distributed, parallel and grid network options. Connectivity protocols include and are not limited to PPP, http, TCP/IP, and ftp over multiple platforms.

FIG.6A is a diagrammatic representation showing the relationship of the intelligent object to an external intelligent object handler ("IOH"), its components and access interfaces, the legacy domain of existing data content, applications, and devices, and an external intelligent object pool (IOP) 204.

In FIG. 6A, unbroken lines ending with arrows on each end 490 represent bi-directional communication between exemplary property panes, components and access interfaces. Dashed lines ending with arrows on each end 492 represent bi-directional communication between optional property panes, components and access interfaces. Crossed lines do not represent connections.

FIG. 6B is a diagrammatic representation of Intelligent Objects providing a conceptual overview of interactive content routing for multiple dimensions of (or “multidimensional”) direct information interchange within and between Intelligent Objects.

With further reference to FIG. 6B, each property pane provides an overview of certain properties of the comprised data and its relationships. For example, property panes describe Intelligent Object ownership and activity history, but also complex, multiple relationships to other data and applications. Vectorized data content information and results of data content comparison and analysis, data annotation, text references, validation status and the like may be flexibly interconnected in a functional manner via these panes and their related components and access interfaces. User viewing and interactivity to define or refine (without writing to the data content) Intelligent Object property pane content presentation and relationship connectivity for new queries, customization and the like takes place through property panes presented at the unified presentation layer.

FIG. 7A is a diagrammatic illustration depicting aspects and embodiments of the invention including exemplary embodiments comprising a Unified Presentation Layer, modules comprising components and access interfaces, and their relationships; as well as the relationship of the Intelligent Object Pool (“IOP”) to its architectural elements, such as for example an external **Intelligent Object Handler 202** and/or Handlers (IOH) **202** and to their components, access interfaces and users, to Intelligent Objects (“IMOs”), and via the **Intelligent Object Handler 202** and Intelligent Objects to the external “Legacy” domain of existing heterogeneous data content or databases, applications, systems and/or devices.

With further reference to the embodiment in FIG. 7A, unbroken lines ending with arrows on each end 490 represent bi-directional communication between exemplary property panes, components and access interfaces. Dashed lines ending with arrows on each end 492 represent bi-directional communication between optional property panes, components and access interfaces. (Generally, crossed lines 491 do not represent connections.)

FIG. 7B is a diagrammatic representation of an intelligent object pool (IOP) comprising intelligent objects (IMO) 200 and a data subset intra-Pool (iPool) 155 of intelligent objects defined according to ownership, similarity or other boundary definitions.

FIG. 8 is a diagrammatic representation of an exemplary embodiment of the claimed elements and their relationships, showing the relationship of the Intelligent Object Handler to Intelligent Objects, comprised components and access interfaces, the legacy domain of existing data content, applications, and devices, and an external Intelligent Object Pool ("IOP").

In FIG. 8, unbroken lines ending with arrows on each end 490 represent bi-directional communication between exemplary property panes, components and access interfaces. Dashed lines ending with arrows on each end 492 represent bi-directional communication between optional property panes, components and access interfaces. (Generally, crossed lines 491 do not represent connections.)

With reference to FIG. 9, in a global, heterogeneous environment, data from diversified sources are functionally integrated via intelligent objects and their intelligent object pool (IOP) 204, utilizing automated access definition protocols, multiple alias integration and addressing, data field mapping and state management to allow for integrated data access, analysis and exchange utilizing distributed data content presented within the Client iPool.

The chart in FIG. 9 depicts user relationships to the data on database and iPool levels. In the figure, the upper section of the diagram (i.e. "Client iPool", "Restricted", "Client Data", "Resources") depicts data within intranets, LANs and the like, while the lower, larger part of the diagram contains several different forms of public and/or otherwise web-accessible data sources.

The depiction below in FIG. 10 represents a process model describing interactions of modules for an exemplary embodiment of the intelligent object pool (IOP) 204.

In FIG. 10, data from a global data resource are passed through an access interface consisting of a security layer 404, a set of access and/or exchange protocols 406 and integrity assessment procedures 408, to the intelligent objects comprised within the intelligent object pool 204 for unified queries and analyses. The pathways involved in unified direct intelligent object-to-intelligent object 455-456 and iPool-to-iPool 457-458 data interaction generate the real-time answer and pass it back through processes including aggregate meta-data indexing 422, significance generation and aggregation 473-476 and security 404 to the intelligent object handler 202.

The depiction below in FIG. 11 is a representation of the graphical user interface window within the unified presentation layer displaying algorithms included for clustering of intelligent

object data. A number of property panes showing dendrograms results are seen behind the clustering algorithms menu in FIG. 11.

The depiction below in FIG. 12 shows an interface iPool Zoom viewer for viewing iPool data relationships, utilizing techniques that include dendrograms and self-organizing maps (SOM).

Attributes and interactions between individual intelligent object (IMO) data within the set of objects are outlined in the margins of this exemplary display, FIG. 12.

The depiction below in FIG. 13 shows an iPool Zoom interface for viewing iPool data relationships, utilizing Principal Components Analysis (PCA).

In one exemplary and advantageous embodiment the Intelligent Object Pool **(IOP) 204**, comprised by an information technology platform **(ITP) 151** (such as for example, the Sentient IT Platform version 1.0.4beta, made by Biosentients, Inc. of 1325 61st Street, Emeryville, CA 94608-2117) advantageously enabled in software, provides an organizational architecture and framework for Intelligent Object (IMO) data architecture. Core architectural elements of the information technology platform 151 include the **Intelligent Object Pool 204**, an external **Intelligent Object Handler 202,** and **Intelligent Objects 200**, each comprising some combination of components, access interfaces, some or all coupled or otherwise interacting via included methods and processing routines and subroutines. In some embodiments, intra-Pools (iPools) **155**, comprising subsets of Intelligent Object Data within the **Intelligent Object Pool 204** data pool may also be considered as a core architectural element. Some functionally meaningful combinations of components and / or access interfaces may also be defined as stand-alone modules or as customizable plug-ins to other architectures or systems. Other optional elements may also be provided within or coupled with the core architectural elements.

Intelligent Object Pool 204

With reference to an embodiment illustrated in FIG. 7A, the **Intelligent Object Pool 204** comprises Intelligent Objects, representing the global entity (or portion subset of global entity called an intra-Pool or iPool **155**) of accessible **Intelligent Object 200** data content presented within a functionally integrated Unified Presentation Layer **206**. In addition to interaction with the **Intelligent Objects 200**, the **Intelligent Object Pool 204** may be activated and directed by the **Intelligent Object Handler** 202. It is noted that in FIG. 7A, unbroken lines ending with arrows on each end represent bi-directional communication between exemplary property panes,

components and access interfaces; dashed lines ending with arrows on each end represent bi-directional communication between optional property panes, components and access interfaces; and lines that may cross do not represent connections.

The Intelligent Object Pool 204, Intelligent Object Handler 202 and Intelligent Objects (IMO) 200 comprise a unified presentation layer; components comprising defined methods, procedures, and processes; and access interfaces. The access interfaces provide unique conventions used to allow communication between components and in some cases other interfaces, for direct linking of vectorized pointers to accessible Intelligent Objects (IMO) 200 data content, for direct linking of meta-data; and for linking and activation of component information and processes.

Other optional but advantageous data-enabling methods and processes may for example include one or more of the following, such as for example the customizable exchange of various combinations of information required for queries and commands within heterogeneous data and applications environments, and modules for such as automated distributed learning (DLE) and knowledge extraction (KEE).

The inventive information technology platform architecture utilizes Intelligent Object data content properties and values for core functionality, such as for example, aggregation, organization and other processing of results of vectorized data content analysis, or other information interchanged via components and interfaces, such as dynamically definable meta-data definitions.

With reference to FIG. 6A a raw data matrix property pane (RDM) 1042 included within Intelligent Objects (IMO) defines data content addressing and provides vectorized access information for linking of diverse types of data content. Additionally, the matrix structure descriptor (MSD) 1046 provides processes required by the application / database definition router (ADDR) 1028 and data link insertion component (DLI) 1044 for data field mapping and gating of vectorized access to individual data fields between objects, applications and databases. Examples of exemplary enabling code for the raw data matrix property pane (RDM) 1042 and matrix structure descriptor (MSD) 1046 interface are provided in a concurrently filed copending United States Utility Patent Application Serial No. 09/___,___ (Attorney Docket No. A-70135/RMA) entitled and describing among other subject matter embodiments of an Intelligent Object (IMO) 200 application.

Vectorized access, routing, translation, linking and comparison of data content include, for example data content information such as RGB values or data field contents such as literature

reference pointers and data content subsets ("workspaces") defined to granularity from entire data files to data content subsets as small as single byte workspaces, such as a single band, spot or a set of bands or spots in a gel electrophoresis pattern. Direct interchange of information between data objects and between data objects, heterogeneous applications and a various combinations of components, access interfaces, and presentation layers includes processing information, commands, queries, meta-data, data content and said defined workspace subsets. Meta-data includes, for example, alphanumeric information defining the data content and its relationships, definitions for linking Intelligent Objects according to their detected functional or semantically defined relationships, and lookup tables for data content definition and presentation information.

Comprised meta-data index information may be utilized to optimize linking, organization and direction of the vectorized data content access and routing requests, processing commands, and information interchange between such as Intelligent Objects, heterogeneous and/or homogeneous data objects and data content, applications, data resources and/or databases. An object pane descriptor (OPD) provides an overview of information for any particular Intelligent Object state and its relationships, for easy user review and interactivity with functional linking and vectorized accessing and routing of data content, meta-data and information such as queries, reports and processing results.

A status management component (SMC) within each Intelligent Object (IMO) 200 initiates direct object-to-object interchange of information including data content, meta-data, queries and processing commands in parallel with corresponding Intelligent Objects via their comprised object query interfaces. The status management component (SMC) directs the interactive content routing of specified data vectors and/or meta-data index information between object query interfaces and of other Intelligent Objects for linking, comparison and relationship definition according to comprised algorithms described and defined elsewhere, and relays the data content information to the external object state engine (OSE) and result aggregation engine interfaces (RAEi) within the Intelligent Object Pool (IOP) 204.

Direct information interchange (DII) may include various methods, procedures, and processes that provide such functionality such as one or more of: (a) activity synchronization, (b) generation of authentication and availability states, (c) verification of integrity states for quality assurance, (d) maintenance of exchange and ownership histories, and (e) status updating.

Activities may for example include one or more of the following: fielding and permission for access privileges; recordation and reporting of royalties or fees for data exchange; including recordation of detail including data attributes accessed, requests for linking of data attributes;

assessment of Intelligent Object (IMO) 200 data integrity, of the integrity of the entire Intelligent Object Pool contents, and/or of iPool data subsets; or other activities provided by the system.

In one embodiment, activity synchronization states may include for example information required for synchronizing two or more processes that depend upon the simultaneous or closely linked occurrence of specific related events in time, or that comprise closely related properties or functions, such as for example: synchronized linking for common responses to queries; commands for window scrolling linked across data; requests for batch information updating; automated batch information updating; linking to common timing signals; and the like. Activity synchronization states may be required by and provided to various combinations of Intelligent Objects, components and access interfaces and are generated, for example, by the object state engine (OSE) 208 within the Intelligent Object Handler (IOH) 202, interacting with the status management component (SMC) 1016 of the Intelligent Molecular Object (IMO) 200. Activity synchronization may also be generated by additional components and interfaces such as are comprised within the Intelligent Object Pool (IOP) 204 for synchronization of meta-data linking (RML) and (RMLi), and for synchronizing iPool-to-iPool queries (PPQ) and (PPQi).

Availability states may include for example, on-line or offline status; ranked or otherwise sorted aliases and / or multiple addresses for data content access; validation ranking to determine availability of data for such as regulatory filing requests; and the like. Availability states are provided to the comprised iPool Availability Monitor by external components and access interfaces, such as for example the object access manager (OAM) 1036 comprised within the Intelligent Object (IMO) 200; the application / database definition generator (ADG) and legacy synchronization interface (LSI) 2018 comprised within the Intelligent Object Handler (IOH) 202; and the iPool availability monitor (iPAM) and (iPAMi) comprised within the Intelligent Object Pool (IOP) 204.

Authentication states may include for example, the status of verification regarding the unique identity of an Intelligent Object, a person, a machine, a session or a process; the access or availability status of the uniquely identified Intelligent Object, person, machine, or process; verification that information, queries or commands really come from their stated source; and the like. The generation of authentication states involves processes comprised within the Intelligent Object Handler (IOH) 202 such as for user definition and administration (UDA) 2000; within the Intelligent Object (IMO) 200 such as for unique identification, accessing and routing (1006-1012) and status management (SMC) 1016; and within the Intelligent Object Pool (IOP), such as for the iPool security authentication module comprising iPSA and iPSAi .

Integrity states for quality assurance may for example include or be selected from one or more of the following integrity states: data integrity verified; data integrity altered; alerts regarding Intelligent Object integrity violation status; alerts regarding processes that may or will violate Intelligent Object integrity; activation of an interactive user interface providing a description of the alert; provision of optional processing methods and/or cancellation of the processes in case of integrity violation alerts; and the like. Verification of these integrity states for quality assurance involves processes comprised within the Intelligent Object Handler (IOH) 202 such generated by the object state engine (OSE) 208 and legacy synchronization interface (LSI) 2018; within the Intelligent Object (IMO) 200 such as for status management (SMC) 1016 and raw data matrix definition (RDM) 1042; and within the Intelligent Object Pool (IOP), such as for Intelligent Object integrity assessment (OIA) and (OIAi), and iPool integrity assessment (iPIA) and (iPIAi).

Exchange and ownership histories include for example histories chronicling: Intelligent Object and iPool ownership status; Intelligent Object and iPool ownership privileges and parameters for exchange; recorded criteria for exchange such as licensing, subscription, royalty or fee agreements; histories of Intelligent Object data attribute access and attribute linking; calculated exchange charges, fees or royalties; or other records of exchange accounting. Generation and maintenance of exchange and ownership histories involves processes comprised within the Intelligent Object Handler (IOH) 202 such as for example, the user definition and administration (UDA) 2000 component and object state engine (OSE) 208; within the Intelligent Object (IMO) 200 such as for unique identification and (UID) 1006 and (UIDi) 1012; and within the Intelligent Object Pool (IOP), such as for implementing and monitoring exchange protocols (iPEP) and (iPEPi).

Additionally the **Intelligent Object Pool 204** comprises methods, procedures, and processes for one or more of the following: fielding queries, iPool 155 boundary management, iPool data integrity management, data exchange management, data integrity assessment, data definition, query and processing optimization, data and data subset management, significance generation, result aggregation, and data relationship viewing. These methods, procedures, or processes are activated and implemented according to predetermined or dynamically determined rules, policies, and procedures. The rules, policies, and procedures may be any one or more of automated, event-driven, interactive, and/or user-directed, secure, audited, and validated.

Methods for result aggregation (RAE), and (RAEi) are generally provided in a minimal embodiment of the invention, while other methods including for example exchange protocols

(iPEP, (iPEPi); meta-data indices methods (PPQ), and (PPQi) ; and / or pool content access methods (DLE), and (DLEi) are optional but advantageous in most cases. Additionally, stand-alone functional modules constructed out of the defined components and access interfaces may be assembled to provide customized functionality.

With further reference to the embodiment of the Intelligent Object Pool 202 in FIG. 7B, at a top-level, the **Intelligent Object Pool 204** comprises methods, protocols, and definitions that enable the functionality and utilization of the **Intelligent Object Pool 204** and data subset **iPools 155**. Recall that, in at least one embodiment of the invention, **iPools 155** describe any subset of Intelligent Object data comprised within an Intelligent Object Pool 204. The iPools may be flexibly defined by the user according to a variety of parameters including ownership, data attributes, experimental design, and the like. The protocols, definitions, and methods include Pool boundary protocols **402**, meta-data indices indices **410** and definitions **411**, and pool content access methods and processes **420**, as represented in FIG. 7A.

In one particular embodiment, an information technology platform provides an object pool enabled in computer program software for any object-oriented data architecture so that the object pool architecture and methodology are not limited only to the particular intelligent objects or intelligent molecular objects specifically described elsewhere herein, but are merely exemplary.

In other embodiments, additional modules, components, and interfaces for a one or more optional but advantageous distributed learning engines and/or knowledge extraction are provided. In the description that follow and that are provided elsewhere in this specification, the terms modules, components, and interfaces are used to describe aspects of embodiments of the invention. While some differences may generally exist between these modules, components, and interfaces as the terms apply to difference aspects of the inventive structure, method, and computer program and computer program product, the following guidelines (though not definitions) in understanding these phrases is provided.

A module is a top-level definition that defines an instance of indicated functionality which may or may not be required and provided for a specific user installation. Modules may be compiled separately and a module's implementation can be changed without requiring changes to other components, interfaces, or modules. A module usually includes a component and an interface or some combination of these, and may be designated by a "m" suffix following the module acronym (such as RAE for the Results Aggregation Engine component or RAEm for the Results Aggregation Engine module). Components provide core processing activity required for

given functionality, comprising methods generic to the overall architecture for advertising and enabling its functionality, whether via interactive methods such procedures, routines, or information interchange with some combination of components and / or access interfaces. An interface provides unique conventions used to allow communication between components and in some cases other interfaces, for direct linking of vectorized pointers to accessible Intelligent Objects (IMO) 200 data content, for linking and activation of meta-data; and for linking and activation of component information and processes, and may be designated by a "i" suffix following the module acronym (such as RAEi for the Results Aggregation Engine interface). It will be appreciated that as each module, component, and interface is different, reference should be made to the detailed descriptions of each particular module, component, and interface to ascertain its structure, function, and operation.

Also, for purposed of clarity, certain modules, components and access interfaces may be grouped into functionally related sets, such as Pool boundary protocols (PBP), Meta-Data Indices (MDX), and Pool Content Access (PCA) definitions and methods.

Pool boundary protocols (PBP) 402

In one aspect, the pool boundary protocols **(PBP) 402** are primarily responsible for providing data security, providing and enabling data exchange, and providing data integrity assessment.

The Pool boundary protocol **(PBP) 402** methods and definitions described herein generally provide (i) pool security protocols and procedures **404** that provide data security, (ii) pool boundary exchange protocols and procedures **406** that provide definitions for data ownership, royalty agreements, data access fee agreements and enable detailed use-tracking and reporting to enable accurate charging of fees, auditing of revenue, and other detailed assessment and control for access and exchange of data subsets, or **iPools** 155, and (iii) Pool integrity protocols and procedures **408** that provide and maintain data integrity assessment for individual data objects as wells as data subset **iPools** 155. Recall that the intelligent object data includes data content stored in global and/or local data resources of potentially heterogeneous type and/or structure. In one embodiment, the pool boundary protocols **(PBP) 402** include a plurality of functional modules comprising components and access interfaces as described in greater detail elsewhere herein.

In one embodiment, a set of **pool boundary protocols (PBP) 402** are provided, comprising a plurality of modules: (i) an iPool Security Authentication module (IPSAm), (ii) an

iPool Availability Monitor module (IPAMm), (iii) an iPool Exchange Protocol module (IPEPm), (iv) an Object integrity assessment module (OIAm), and (v) an iPool Integrity Assessment module (IPIAm).

The iPool Security Authentication module (IPSAm) authenticates and permits or rejects iPool data requests according to user login and object data identification, and in the embodiment of FIG. 7A comprises an iPool security authentication component **(iPSAc) 431**, and an iPool security authentication interface **(iPSAi) 432**).

The iPool Availability Monitor module (IPAMm) assures availability of iPool data content, and in the embodiment of FIG. 7A comprises an iPool Availability Monitor component **(iPAMc) 433**, and an iPool Availability Monitor interface **(iPAMi) 434**).

The iPool Exchange Protocol module (IPEPm) governs and manages data ownership and exchange, and in the embodiment of FIG. 7A comprises an iPool Exchange Protocol component **(iPEPc) 435**, and iPool Exchange Protocol interface **(iPEPi) 436**.

The Object integrity assessment module (OIAm) assesses object integrity for auditing, security, validation, and quality assurance/quality control; and in the embodiment of FIG. 7A comprises an Object integrity assessment component (OIAc) **437**, and an Object integrity assessment interface (OIAi) **438**.

Finally, the iPool Integrity Assessment module (IPIAm) assesses data integrity within a defined iPool for security and quality assurance and quality control and which provides alerts regarding iPool validation status, and in the embodiment of FIG. 7A comprises an iPool Integrity Assessment component (IPIAc) **439**, and an iPool Integrity Assessment interface (IPIAi) **440**, which assesses data integrity within a defined iPool for security and quality assurance and quality control and which provides alerts regarding iPool validation status.

Where the functionality provided by a particular module is not required for a particular application, its provision is optional to the IOP 204 and not required by these embodiments of the invention. Furthermore, additional modules such as modules comprising components 441, 443, and/or interfaces 442, 444 may optionally be provided as the system, architecture, topology, and methodology are adaptable and expandable to add additional capabilities as the need arises.

In an exemplary embodiment, a set of Pool boundary protocols 402 provided within the Intelligent Object Pool 204 includes a set or sets of computer program software instructions enabling unidirectional and preferably bi-directional information interchange with other system components and access interfaces including but not limited to components and interfaces associated with the external **Intelligent Object Handler 202** (IOH), Intelligent Object Pool

(IOP) 204, **iPools 155**, and Intelligent Objects including Intelligent Molecular Objects (IMO) **200**. Recall that although this description focuses on particular embodiment of the IMO and IOH as described elsewhere herein, the structure and methods described relative to the IOP and components of the IOP are applicable to other objects and object handlers.

Defined data subsets or intra-Pools (iPools) **155**, comprise subsets of Intelligent Object data within the Intelligent Object Pool 204. When the user requests access to the entire available data included within the Intelligent Object Pool, or is accessing the entire content of their Intelligent Object Pool for integrity verification, the terms data pool, or simply pool may be used to describe the entity of Intelligent Object data and their contents included within the Intelligent Object Pool database structure. Normally, however, the term iPool is used to describe any set of data which has been defined, recognizing that even all data comprised within the Intelligent Object Pool database structure does not the define the global entity of data, but is rather a subset (or intra-Pool) of the global data resource, or global data pool. In most cases, iPool is used to define subsets of data delimited within the Intelligent Object Pool 204 by Pool boundary protocols such as ownership, conditions for access or exchange, data interdependencies and content-based relationships, and the like.

Pool boundary protocols, Meta-data Indices, and Pool Content Access definitions provide descriptive headings referring to groupings of included components and access interfaces within a functional block of the Intelligent Object Pool 204. The Pool boundary protocols, for example, comprise components and access interfaces, which together may comprise functional modules for Pool boundary protocols, definitions and activities and iPool boundary protocols, definitions and activities. For example, the combination of iPSA and iPSAi comprise a Pool boundary protocol module for security. In analogous manner, the combination of iPAM with iPAMi comprise a Pool boundary protocol module for security and exchange.

The pool boundary protocols (PBP) comprise modules, interfaces, components, methods and processes which enable data subset pools or iPool 156 boundary definitions and user interactivity with these data subset iPools 156. Generally, Pool boundary protocols (PBP) provide access control, tracking, recordation and management of data exchange, and data set integrity assessment required for iPool subset 156 boundaries and parameters . In particular, Pool boundary protocols allow for unified definition and management of data subsets via included component and access interface for (a) access and security; (b) integrity assessment and verification for Intelligent Objects **200** and **iPools 155**; (c) management of data content access for exchange and auditing; (d) access and routing of global and/or local Intelligent Object data; (e)

data source availability management and provision of the information to components and access interfaces as required for state management of **Intelligent Objects 200**, **iPools 155**, and data content information interchange.

Additionally, Pool boundary protocols may be utilized to provide the same or similar functionality for the entire entity of data comprised within the Intelligent Object Pool 202 database structure, in which case the entity of data may be called a data Pool or Pool. Because iPools 156 are subsets of the full data "Pool", iPools may be flexibly defined according to the properties of comprised data in relationship to access, availability, ownership, integrity, and the like.

In one embodiment, the Pool boundary protocols desirably and advantageously utilize object-oriented data structures, while other embodiments use other than object-oriented data structures. The use of object-orientated data structures is advantageous in some information technology platform implementations, including the Sentient Platform described in detail herein elsewhere, in part because of the broad acceptance and existence heterogeneous object-oriented data structures currently, and also because of the existence of certain valuable data annotation, meta-data, or other data-enabling techniques provided by the object-oriented data structure. On the other hand, non-object-oriented data structures may also be used in part because of the broad acceptance and existence heterogeneous non-object data currently, and also because the technologies defined herein are ideally suited to accessing and defining non-object data content, and to translating the data content for presentation within a variety of environments, both object-oriented and non-object-oriented.

iPool Security Authentication Component (iPSAc) 431

The iPool Security Authentication component **(iPSAc) 431** is primarily responsible for permitting or denying a request for access to features and capabilities of the system and modules, components, and interfaces of the system, and/or to contents of the system; optionally including a level of access when any such access is permitted.

In an exemplary embodiment, an iPool Security Authentication component **(iPSAc) 431** contains a set of Security Authentication software instructions comprising Pool boundary protocols, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following: Unified Presentation Layer; iPool Security Authentication interface; and Pool boundary protocol components. Additionally, a set of comprised definitions provide automated and/or interactive methods and

processes including definition of iPool access permissions; definition of iPool ownership and privileges and criteria for permission or denial of specific access, routing and processing activities; including but not limited to permission or denial to access and/or route subsets of data content defined to the level of single bytes by activation of vectorized pointers to the data content. Additionally, the iPool Security Authentication component desirably includes methods and processes including permission or denial to access and/or route iPool meta-data information and definitions; including but not limited to iPool content overviews; and iPool relationship information. Furthermore, included methods and processes desirably include automated and/or interactive authentication and permission or denial of iPool access requests according to correspondence of user login information; user definition and administration permissions provided by an interface to an external **Intelligent Object Handler 202**; iPool access permissions; and comprised Intelligent Object data identification.

In an exemplary embodiment, the iPool Security Authentication component **(iPSAc) 431** comprises a iPool Security Authentication module (IPSAm) within an information technology platform **151** for Intelligent Objects. In another exemplary embodiment, the iPool Security Authentication module (IPSAm) optionally but advantageously comprises a module within an information technology platform using object-oriented data structures.

__iPool Security Authentication Interface (iPSAi) 432__

In one aspect, the iPool Security Authentication interface **(iPSAi) 432** is primarily responsible for enabling detection and extraction of information required for security authentication by the iPool Security Authentication component **(iPSAc) 431** from intelligent objects and their data content, the IOP 204, iPools 155, and components and interfaces thereof. It may also be responsible for routing information derived by iPSAc 431 directly to IOP 204, iPools 155, and components and interfaces thereof.

Additionally, methods and processes are provided, which route information derived from the iPool Security Authentication component directly to the **Intelligent Object Pool 204**; **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces; provision of iPool Security Authentication component activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity; recordation of the component information

interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, an iPool Security Authentication interface **(iPSAi) 432** includes a set of Security Authentication interface computer software code instructions including Pool boundary protocols, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to one, more, or all of the following: an iPool Security Authentication component; one or more Intelligent Object Pool access interfaces; an **Intelligent Object Handler 202** (IOH); and **Intelligent Molecular Objects 200** (IOM).

Information detection and extraction methods, procedures, and processes are provided, which enable detection and extraction of information required for iPool Security Authentication, provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204** ; **iPools 155**; components and access interfaces. The information includes but is not limited to one, more, or all of: Intelligent Object data content to be accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information; relationship information; protocols; ontologies; annotations; and/or other data-enabling information; provided directly to the iPool Security Authentication component.

Additionally, methods and processes may be provided, which route information derived from the iPool Security Authentication component directly (preferred) or indirectly to the **Intelligent Object Pool 204**; **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204, iPools 155**, components and interfaces; provision of iPool Security Authentication component activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity; recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, the iPool Security Authentication interface (iPSAi) is comprised as a module within an information technology platform for Intelligent Objects. In another embodiment, the iPool Security Authentication interface comprises a module within an information technology platform using object-oriented data structures.

iPool Availability Monitoring Component (iPAMc) 433

The iPool Availability Monitoring component **(iPAMc) 433** is primarily responsible for one or more of the following: monitoring availability of iPool data content, managing access for iPool data content, managing alias nomenclature, ranking and managing multiple addressing, maintaining persistent meta-data definition availability, and data and data subset cache definitions for data. Depending upon system configuration and desired capabilities one, several, or all of these responsibilities may be optional. It may also optionally be responsible for data content backup for specified or often used data and multiple-content addressing for specified or often used data.

In an exemplary embodiment, an iPool Availability Monitoring component **(iPAMc) 433** contains or comprises a set of software instructions for Pool boundary protocols, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Unified Presentation Layer; iPool Availability Monitoring interface; and Pool boundary protocol components. A set of Pool boundary protocol definitions are included, which provide automated and/or interactive methods and processes including availability monitoring for iPool data content; access management for iPool data content; nomenclature alias management; multiple addressing ranking and management; maintenance of persistent meta-data definition availability; and data and data subset cache definitions for data including but not limited to specified Intelligent Object data; Intelligent Object data content; offline Intelligent Object data; and for data including but not limited to user defined; often used; and recently used. An additional set of Pool boundary protocol definitions may be provided which enable or provide automated and/or interactive methods and processes including data content backups for specified or often used data and multiple-content addressing for specified or often used data.

In an exemplary embodiment, the iPool Availability Monitoring component **(iPAMc) 433** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the iPool Availability Monitoring component comprised as a module within an information technology platform using object-oriented data structures.

iPool Availability Monitoring Interface (iPAMi) 434

The iPool Availability Monitoring interface **(iPAMi) 434** is primarily responsible for detecting and extracting direct information required for availability monitoring, such as may be provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155,

and/or other components, and/or interfaces. The iPAMi 434 may also be responsible for routing information derived from the iPool Availability Monitoring component directly to the Intelligent Object Pool 204, iPools 155, and components and access interfaces. It may further optionally be responsible for providing the iPool Availability Monitoring component activity information to access interfaces and components as required for utility. The utility desired may for example include one or more of: synchronization of component information interchange, activity recordation of both the component information interchange and activity, auditing of the component information interchange and activity, validation of the component information interchange and activity, or other utility that may be desired and provided within the system and method.

In an exemplary embodiment, an iPool Availability Monitoring interface **(iPAMi) 434** contains a set of software instructions comprising Pool boundary protocols, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following iPool Availability Monitoring component; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202** and Intelligent Objects. Methods and processes are provided, which detect and extract direct information required for availability monitoring, provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information; relationship information; protocols; ontologies; annotations and other data-enabling information; provided directly to the iPool Availability Monitoring component.

Additionally, methods and processes are provided, which route information derived from the iPool Availability Monitoring component directly to the **Intelligent Object Pool 204**; **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of the iPool Availability Monitoring component activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary aspect, the iPool Availability Monitoring interface **(iPAMi) 434** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the iPool Availability Monitoring interface comprised as a module within an information technology platform using object-oriented data structures.

iPool Exchange Protocol Component (iPEPc) 435

The iPool Exchange Protocol Component (iPEPc) 435 is primarily responsible for defining iPool ownership status; defining iPool ownership privileges and parameters; and for providing detailed definitions regarding criteria for data and/or data attribute or other information exchange. The iPEPc 435 may also be responsible for managing user ownership, user ownership recordation, authentication and permission or denial of iPool access requests, and other administrative functions.

In an exemplary embodiment, an iPool Exchange Protocol component **(iPEPc) 435** contains a set of software instructions comprising Pool boundary protocols, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Unified Presentation Layer; iPool Exchange Protocol interface; and Pool boundary protocol components. A comprised set of Pool boundary protocol definitions provide automated and/or interactive methods and processes including secure and interactive definition of iPool ownership status; definition of iPool ownership privileges and parameters; and detailed definition regarding criteria for exchange, including protocols and/or algorithms for Intelligent Object data attribute access; protocols and/or algorithms for such as ownership, licensing, and subscriptions; protocols and/or algorithms for such as exchange charges, and royalties; protocols and/or algorithms for exchange accounting; and account definitions. Additionally, a set of Pool boundary protocol definitions are comprised which provide automated and/or interactive methods and processes including user ownership management user ownership recordation; authentication and permission or denial of iPool access requests; according to correspondence of user ownership verification to comprised definitions such as for access, routing and processing; user exchange activity history provided by an interface to an external **Intelligent Object Handler 202**; data attribute access and linking activities; ownership, licensing, and subscription status; fees, charges and royalty status; including previewing of fees, charges and/or royalty terms and conditions according to proposed queries and/or user commands.

In an exemplary embodiment, the iPool Exchange Protocol component **(iPEPc) 435** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the iPool Exchange Protocol component is optionally but advantageously comprised as a module within an information technology platform using object-oriented data structures.

iPool Exchange Protocol Interface (iPEPi) 436

The iPool Exchange Protocol Interface (iPEPi) 436 is primarily responsible for detecting and extracting direct information required for exchange protocols, provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155, and various components and access interfaces. The iPool Exchange Protocol Interface (iPEPi) 436 may additionally be responsible for routing information derived from the iPool Exchange Protocol component directly to the Intelligent Object, **Intelligent Object Handler 202**, Pool **154** and **iPools 155**, components and access interfaces. It may further optionally be responsible for providing access and/or other control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. It may additionally be responsible for providing iPool Exchange Protocol component activity information to access interfaces and components as required for utility. The utility required or otherwise provided it not required, may for example include one or more of the following: synchronization of the component information interchange, activity recordation of the component information interchange and activity, auditing of the component information interchange and activity, validation of the component information interchange and activity, and combinations thereof.

In an exemplary embodiment, an iPool Exchange Protocol interface **(iPEPi) 436** contains a set of software instructions comprising Pool boundary protocols, methods and processes including bi-directional information interchange with components and access interfaces of and/or including but not limited to the following iPool Exchange Protocol component; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202**; and **Intelligent Molecular Objects 200**. Methods and processes are provided, which detect and extract direct information required for exchange protocols, provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including one or more of Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information relationship information; protocols; ontologies; annotations; and other data-enabling information; provided directly to the iPool Exchange Protocol component.

Additionally, methods and processes are provided, which route information derived from the iPool Exchange Protocol component directly to the Intelligent Object, **Intelligent Object Handler 202**, Pool **154** and **iPools 155**, components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204, iPools 155**, components and interfaces. Comprised methods and processes also include provision of iPool Exchange Protocol component activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, the iPool Exchange Protocol interface **(iPEPi) 436** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the iPool Exchange Protocol interface optionally comprises a module within an information technology platform using object-oriented data structures.

Object Integrity Assessment Component (OIAc) 437

The Object Integrity Assessment Component (OIAc) 437 is primarily responsible for the application of comprised methods and processes to Intelligent Object state histories and/or data content status, such as for example data checksums; and review of Intelligent Object state histories in order to provide one or more of auditing, security, validation, and quality assurance/quality control. It may also be responsible for providing alerts regarding Intelligent Object integrity violation status and alerts regarding processes that may or will violate Intelligent Object integrity, and optionally for activating an interface for the alert and optionally providing processing methods and/or cancellation of the processes for which the alert was generated.

In an exemplary embodiment, an **object integrity assessment component (OIAc) 437** contains a set of software instructions comprising pool boundary protocols methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following unified presentation layer; iPool Integrity Assessment interface; and other Pool boundary protocol components. A set of Pool boundary protocol definitions are included, which provide automated and/or interactive methods and processes including assessment of Intelligent Object and data content integrity based on techniques including but not limited to iPool checksums; and application of comprised methods and processes to iPool

histories and status, including auditing; security; validation; and quality assurance/quality control. Additionally, a set of Pool boundary protocol definitions are included, which provide automated and/or interactive methods and processes including alerts regarding iPool integrity violation status; alerts regarding processes that may or will violate iPool integrity; and activation of an interactive user interface providing a description of the alert and providing optional processing methods and/or cancellation of the processes.

In an exemplary aspect, the Object integrity assessment component comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the Object integrity assessment component comprised as a module within an information technology platform using object-oriented data structures.

<u>Object Integrity Assessment Interface (OIAi) 438</u>

The Object Integrity Assessment Interface (OIAi) 438 is primarily responsible for detecting and extracting direct information required for object integrity assessment provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155 and components and access interfaces; where the information may include but not be limited to Intelligent Object data content accessed via vector pointers, meta-data indices, meta-data for content information, attribute information, relationship information, protocols, ontologies, annotations, and other data-enabling information provided to the Object integrity assessment component. It may also be responsible for routing information derived from the Object integrity assessment component directly to the Intelligent Object, Intelligent Object Handler 202, Pool and iPools 155, components and access interfaces. It may further be responsible for providing access control for information interchange to and from the Intelligent Objects, Intelligent Object Pool 204, iPools 155, components and interfaces. It may additionally be responsible for providing object integrity assessment component activity information to access interfaces and components as required for utility. Such utility may for example include but not be limited to synchronization of the component information interchange, and activity recordation of the component information interchange and activity, auditing of the component information interchange and activity, validation of the component information interchange and activity, and combinations thereof.

In an exemplary embodiment, an Object integrity assessment interface **(OIAi) 438** contains a set of software instructions comprising Pool boundary protocols, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Object integrity assessment component; Intelligent

Object Pool access interfaces; **Intelligent Object Handler 202**; and **Intelligent Molecular Objects 200**. Methods and processes are provided, which detect and extract direct information required for object integrity assessment provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information; relationship information; protocols; ontologies; annotations; and other data-enabling information; provided directly to the Object integrity assessment component.

Additionally, methods and processes are provided, which route information derived from the Object integrity assessment component directly to the Intelligent Object, **Intelligent Object Handler 202**, Pool and **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of Object integrity assessment component activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary aspect, the iPool Object integrity assessment interface **(OIAi) 438** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the Object integrity assessment interface comprised as a module within an information technology platform using object-oriented data structures.

<u>iPool Integrity Assessment Component (iPIAc) 439</u>

The iPool Integrity Assessment Component (iPIAc) 439 is primarily responsible for assessment of Intelligent Object and data content integrity, such as assessing intelligent object and data content integrity using iPool checksums, and iPool related auditing, security, validation, and quality assurance/quality control. It may also be responsible for generating and/or processing alerts regarding iPool integrity violation status, alerts regarding processes that may or will violate iPool integrity, and activation of a user interface for identifying the nature of the alert and presenting options for processing the alert and or the process generating the alert.

In an exemplary embodiment, an iPool Integrity Assessment component 439 contains a set of software instructions comprising Pool Boundary Protocols, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Unified Presentation Layer; iPool Integrity Assessment interface; and Pool Boundary Protocol components. A set of Pool Boundary Protocol definitions are comprised, which provide automated and/or interactive methods and processes including assessment of Intelligent Object and data content integrity based on techniques including but not limited to iPool checksums; and application of comprised methods and processes to iPool histories and status, including auditing; security; validation; and quality assurance/quality control. Additionally, a set of Pool Boundary Protocol definitions are comprised, which provide automated and/or interactive methods and processes including alerts regarding iPool integrity violation status; alerts regarding processes that may or will violate iPool integrity; and activation of an interactive user interface providing a description of the alert and providing optional processing methods and/or cancellation of the processes.

In an exemplary embodiment, the iPool Integrity Assessment component **(iPIAc) 439** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the iPool Integrity Assessment component comprises a module within an information technology platform that optionally uses object-oriented data structures.

<u>iPool Integrity Assessment Interface (iPIAi) 440</u>

The iPool Integrity Assessment Interface (iPIAi) 440 is primarily responsible for detecting and extracting direct information used for iPool integrity assessment provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155, and components and access interfaces, and provided directly to the iPool Integrity Assessment component. The information may include or be selected from (but not limited to) the set of information consisting of Intelligent Object data content accessed via vector pointers, meta-data indices, including meta-data for content information attribute information, relationship information, protocols, ontologies, annotations, and other data-enabling information. The iPool Integrity Assessment Interface (iPIAi) 440 may also be responsible for routing information derived from the iPool Integrity Assessment component directly to the Intelligent Object, Intelligent Object Handler 202, Pool and iPools 155, and components and access interfaces.

In an exemplary embodiment, an iPool Integrity Assessment interface **(iPIAi) 440** contains a set of software instructions comprising Pool boundary protocols, methods and

processes including bi-directional information interchange with components and access interfaces including but not limited to the following iPool Integrity Assessment component; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202**; and **Intelligent Molecular Objects 200**. Methods and processes are provided, which detect and extract direct information required for iPool integrity assessment provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information; relationship information; protocols; ontologies; annotations; and other data-enabling information; provided directly to the iPool Integrity Assessment component.

Additionally, methods and processes are provided, which route information derived from the iPool Integrity Assessment component directly to the Intelligent Object, **Intelligent Object Handler 202**, Pool and **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of iPool Integrity Assessment component activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary aspect, the iPool Integrity Assessment interface **(iPIAi) 440** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the iPool Integrity Assessment interface comprised as a module within an information technology platform using object-oriented data structures.

Meta-data Indices, Methods, Procedures, and Processes

The meta-data indices are primarily responsible for enabling Intelligent Objects to communicate via data-enabling meta-data algorithms.

Embodiments of the meta-data indices and meta-data indices associated methods, procedures, and processes defined herein comprise the components and access interfaces defined and described in detail below, which enable Intelligent Objects to communicate via data-enabling meta-data algorithms, which include for example data object attribute definition, attribute linking,

access optimization, routing optimization and processing optimization. Meta-data indices include modules comprising components and access interfaces for automated, semi-automated, and/or manual or interactive meta-data indexing, dynamic meta-data linking, active object-to-object query meta-data and iPool-to-iPool meta-data indices protocols.

In one embodiment, the Meta-data Indices include one or more of the following modules: an iPool Meta-data Index module (iMDXm) , a Real-time Meta-data Link module (RMLm), an Object-to-Object Query module (OQMm), and an iPool-to-iPool Query module (PPQm).

The iPool Meta-data Index module (iMDXm), comprising an iPool Meta-data Index component **(iMDXc) 451**, and an iPool Meta-data Index interface **(iMDXi) 452**, provides Meta-data definitions for iPool data subsets based on global object meta-data index content such as data field information, data annotation and other existing meta-data information further defined and described below.

The Real-time Meta-data Link component **(RMLc) 453**, provides for relevant data access based on query parameters and Intelligent Object meta-data index based on global object meta-data index content such as data fields, data annotation and other existing meta-data information further defined and described below.

The **Object-to-Object Query component (OQMc) 455**, provides for query optimization and data-enabled parallel processing based on data object intercommunication regarding query parameters and global object meta-data index content such as data fields, data annotation and other existing meta-data information further defined and described below.

The iPool-to-iPool Query component **(PPQc) 457**, provides for data-enabled parallel processing and query optimization based on data object intercommunication regarding query parameters and iPool data intercommunication, global object meta-data index content such as data fields, data annotation and other existing meta-data information provided by and interchanged between iPool Meta-data Indices as further defined and described below.

In an exemplary embodiment, a set of meta-data indices contain sets of software instructions that provide or enable definitions, methods, procedures and/or processes, and include bi-directional information interchange with components and access interfaces including but not limited to an external **Intelligent Object Handler 202**, **Intelligent Object Pool 204**, **iPools 155**, and Intelligent Objects. The meta-data indices definitions interact with the **Intelligent Object Handler 202** and Intelligent Objects to acquire and direct information required for methods and processes including but not limited to Intelligent Object data definition, translation and

integration; integration of analytical processes; and meta-data based optimization of queries and processing at the level of Intelligent Objects; and **iPools 155**; and optimized methods for Intelligent Object-to-Intelligent Object and iPool-to-iPool intercommunication; utilizing Intelligent Object meta-data and aggregated iPool meta-data applied to interactive presorting and exclusion algorithms, object clustering algorithms, meta-data definition and linking modules; and object-to-object and iPool to iPool meta-data interaction definitions. In an optional aspect, the meta-data indices are comprised within an information technology platform using object-oriented data structures that include but are not limited to Intelligent Objects.

iPool Meta-data Index Component (iMDXc) 451

The iPool Meta-data Index Component (iMDXc) 451 is primarily responsible for enabling optimized result aggregation and real-time exclusion of irrelevant object data layers. It may also be responsible for the provision of meta-data definitions for iPool data subsets.

In an exemplary embodiment, an iPool Meta-data Index component **(iMDXc) 451** contains a set of software instructions comprising meta-data indices methods and processes including bi-directional information interchange with components and access interfaces including but not limited to a Unified Presentation Layer; iPool Meta-data Index interface; and meta-data indices components to enable optimized result aggregation and real-time exclusion of irrelevant object data layers. A set of meta-data indices definitions, which provide automated and/or interactive methods and processes including provision of meta-data definitions for iPool data subsets; including but not limited to automatically provided and/or user-defined information regarding data subset content attributes; legacy data type, structure, and access dependencies; data ownership and access information; and data relationship information made available according to available algorithms for determination of such as, but not limited to similarity; association; contiguity; proximity; dependency; functionality; data activity ranking; data significance ranking; data validation ranking; query optimization based on correspondence of query parameters to iPool Meta-data index content; provision of iPool meta-data information required for methods and processes including but not limited to iPool activation, iPool sorting, iPool-to-iPool queries, iPool data presentation and result generation.

In an exemplary embodiment, the iPool Meta-data Index component **(iMDXc) 451** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the iPool Meta-data Index component optionally comprises a module within an information technology platform using object-oriented data structures.

<u>iPool Meta-data Index Interface (iMDXi) 452</u>

The iPool Meta-data Index Interface (iMDXi) 452 is primarily responsible for detecting and extracting direct information required for meta-data indices, provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155, components, and access interfaces. In some embodiments the information includes or is selected from (but is not limited to) Intelligent Object meta-data indices including meta-data for content information, attribute information, relationship information, protocols, ontologies, annotations and other data-enabling information. Typically, the information is provided directly to the iPool Meta-data Index component. It may also be responsible for routing information derived from the iPool Meta-data Index component to the Intelligent Object Pool 204, iPools 155, components and/or access interfaces. It may further be responsible for controlling access for information interchange to and from the Intelligent Objects, Intelligent Object Pool 204, iPools 155, and/or components and interfaces; and/or for provision of the iPool Meta-data Index component activity information to access interfaces and components as required for desired utility.

In an exemplary embodiment, an iPool Meta-data Index interface **(iMDXi) 452** contains a set of software instructions comprising meta-data indices definitions, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following iPool Meta-data Index component; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202** and Intelligent Objects. Methods and processes are provided, which detect and extract direct information required for meta-data indices, provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object meta-data indices; including meta-data for content information; attribute information; relationship information; protocols; ontologies; annotations and other data-enabling information; provided directly to the iPool Meta-data Index component.

Additionally, methods and processes are provided, which route information derived from the iPool Meta-data Index component directly to the **Intelligent Object Pool 204**; **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of the iPool Meta-data Index component activity information to access interfaces and components as required for utility

including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, the iPool Meta-data Index interface **(iMDXi) 452** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the iPool Meta-data Index interface **(iMDXi) 452** optionally comprises a module within an information technology platform using object-oriented data structures.

Real-time Meta-data Link component (RMLc) 453

The Real-time Meta-data Link component (RMLc) 453 is primarily responsible (along with real-time meta-data link interface (RMLi) 454 for linking of the Intelligent Object (IMO) 200 data according to: (i) their own meta-data definitions, (ii) their participation within data subset iPools, and (iii) optionally to other data access and/or applications definitions (e.g. other meta-data) regarding relationships and interdependencies; so that the data is organized automatically according to its characteristics, provides information to each data object regarding its position within the overall data environment, and provides a dynamically and automatically organized environment for optimal accessing and querying of data and optimized, functional interaction between heterogeneous data objects and applications.

The real-time meta-data link component (RMLc) 453 and interface (RMLi) 454 provide the functionality required for linking of the Intelligent Object (IMO) 200 data according to their own meta-data definitions, according to their participation within data subset iPools and according to other useful data access and / or applications definitions regarding relationships and interdependencies commonly understood as meta-data. This provides for automated organization of the data according to its characteristics, provides information to each data object regarding its position within the overall data environment, and provides a dynamically and automatically organized environment for optimal accessing and querying of data and optimized, functional interaction between heterogeneous data objects and applications.

In an exemplary embodiment, a Real-time Meta-data Link component **(RMLc) 453** contains a set of software instructions comprising meta-data indices methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the unified presentation layer; real-time meta-data link interface **(RMLi) 454** and meta-data indices components.

A set of meta-data indices definitions provide automated and/or interactive methods and processes including detection, fielding and organization of meta-data presented via external components, access interfaces, application and user actions including automated and/or user-directed queries; commands and processing requests to provide automated and organized activation of methods and processes corresponding to the meta-data. These meta-data definitions may include validation and integrity rankings; data attributes; annotations; definitions; tables; lists; addressing protocols; access and routing protocols; processing protocols; histories of query parameters presented and provide for optimized provision of relevant data access and activities based on correspondence of the query and/or processing request meta-data parameters to the pre-organized Intelligent Object (200) and iPool linking of said meta-data definitions. A set of methods and processes are advantageously provided which interact with a Real-time Meta-data Link interface, to provide updating of meta-data index linkages for optimization of information interchange between Intelligent Objects (IMO) 200, and between Intelligent Objects (IMO) 200, **iPools 155**, and related applications, components and interfaces. The information interchange is enabled for example as an object state engine (OSE) 208 included within the Intelligent Object Handler (IOH) 202; and a status management component (SMC) 1016 and object query interface (OQI) 1018 comprised within Intelligent Objects (IMO) 200 which provide for direct comparison of data content accessed via vector subset pointers; as well as of meta-data information; and provision of the data content information and meta-data information for updating of the meta-data indices.

In an exemplary embodiment, the Real-time Meta-data Link component **(RMLc) 453** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the Real-time Meta-data Link component **(RMLc) 453** optionally comprises a module within an information technology platform using object-oriented data structures.

<u>Real-time Meta-data Link interface (RMLi) 454</u>

The Real-time Meta-data Link interface (RMLi) 454 is primarily responsible for detecting and extracting direct information required for meta-data linking and automated organization of Intelligent Objects according to criteria provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155, components, access interfaces, and/or specific automated or user-directed queries or commands. It may also be responsible for routing information derived from the Real-time Meta-data Link component to the Intelligent Object Pool 204, iPools 155, and components and access interfaces. It may further be responsible for

controlling access for information interchange to and from the Intelligent Objects, Intelligent Object Pool 204, iPools 155, components and interfaces; and/or for the provision of the Real-time Meta-data Link component activity information to access interfaces and components as required for organization and optimization of processes.

In an exemplary embodiment, an Real-time Meta-data Link interface **(RMLi) 454** contains a set of software instructions comprising meta-data index definitions and methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Real-time Meta-data Link component; other Intelligent Object Pool access interfaces; **Intelligent Object Handler 202** and Intelligent Objects. Methods and processes are provided, which detect and extract direct information required for meta-data linking and automated organization of Intelligent Objects according to criteria provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces, and specific automated and / or user-directed queries or commands. The information required for organization of data and received by the real-time meta-data link interface (RMLi) includes for example Intelligent Object data content accessed via vector pointers; meta-data index definitions including meta-data for content information, attribute information, and other relationship information; accessing, routing and exchange protocols; data ontologies; software routines comprising pre-defined experimental procedures; data annotations and other data-enabling information; and is provided directly to the Real-time Meta-data Link component.

Additionally, methods and processes are provided, which route information derived from the Real-time Meta-data Link component directly to the **Intelligent Object Pool 204**; **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of the Real-time Meta-data Link component activity information to access interfaces and components as required for organization and optimization of processes such as for example synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary aspect, the Real-time Meta-data Link interface **(RMLi) 454** comprised as a module within an information technology platform for Intelligent Objects. In an optional

aspect, the Real-time Meta-data Link interface comprised as a module within an information technology platform using object-oriented data structures.

Intelligent Object-to-Intelligent Object Query component (OQMc) 455

The Intelligent Object-to-Intelligent Object Query component (OQMc) 455 is primarily responsible for detailed mapping and integration of Intelligent Object meta-data and provides the organizational activities needed or desired for the real-time meta-data linking (RML and RMLi), and optionally for other activities. This mapping and integration may further provide for updating meta-data definitions for Intelligent Object(s), to enable components for meta-data linking and query optimization. It may also be responsible for Intelligent Object-to-Intelligent Object meta-data information interchange optimization. The Intelligent Object-to-Intelligent Object Query meta-data interface (OQMi) may also be responsible for providing information required for the real-time meta-data linking (RML) actions including direct meta-data to meta-data linking between objects, and meta-data information interchange corresponding to automated and/or user-directed queries.

In an exemplary embodiment, an **Intelligent Object-to-Intelligent Object Query component (OQMc) 455,** provides a set of software instructions comprising meta-data indices methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Unified Presentation Layer; Intelligent Object-to-Intelligent Object Query interface **(OQMi) 456** and meta-data indices components.

The Object-to-Object Query meta-data index component (OQM) provides detailed mapping and integration of Intelligent Object meta-data and provides the organizational activities required for the real-time meta-data linking (RML and RMLi), as well as for other activities. Methods are comprised acquisition and provision of dynamically updated and/or user-defined meta-data definitions for each comprised Intelligent Object, to enable components for meta-data linking and query optimization; including but not limited to definitions for Intelligent Object state; Intelligent Object data and data content addressing; association; linking; weighting; sorting; ranking of such as attribute; annotation; structure; function; type; matrix definition; field mapping and vector pointers. These meta-data definitions for Intelligent Object-to-Intelligent Object information interchange comprise for example such as previously mentioned validation and integrity rankings; data attributes; annotations; definitions; tables; lists; addressing protocols; access and routing protocols; processing protocols; and histories of query parameters presented. This information may be provided via meta-data comprised within object data comprised within

Intelligent Object data, as well as by various components and access interfaces. Processing is provided for Intelligent Object-to-Intelligent Object meta-data information interchange optimization; comprising organization of the Intelligent Object meta-data linking according to information including but not limited to content attributes; validation state; ranking; relationships; associations; and the like, and for query optimization; based on dynamic updating of the Intelligent Object organization and linking for information interchange; according to correspondence between the information and queries; query histories; commands; command histories; and/or other access, routing and processing actions. Additionally, the Intelligent Object-to-Intelligent Object Query meta-data interface (OQMi) provides information required for the real-time meta-data linking (RML) actions including direct meta-data to meta-data linking between objects; and meta-data information interchange corresponding to automated and/or user-directed queries.

In an exemplary embodiment, the Intelligent Object-to-Intelligent Object Query component **(OQMc) 455** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the Intelligent Object-to-Intelligent Object Query component **(OQMc) 455** comprises a module within an information technology platform optionally using object-oriented data structures.

<u>Intelligent Object-to-Intelligent Object Query interface (OQMi) 456</u>

The Intelligent Object-to-Intelligent Object Query interface (OQMi) 456 is primarily responsible for detecting and extracting direct information desired and/or required for object-to-object queries provided by the Intelligent Objects and their data content; Intelligent Object Pool 204, iPools 155, components, and access interfaces; where the information may include but is not limited to Intelligent Object data content accessed via vector pointers, meta-data indices (including meta-data for content information), attribute information; relationship information, protocols, ontologies, annotations and other data-enabling information. As with several of the other inventive interfaces, this information may advantageously be provided directly, in this case directly to the Intelligent Object-to-Intelligent Object Query component. It may also be responsible for routing information derived from the Intelligent Object-to-Intelligent Object Query component to the Intelligent Object Pool 204, iPools 155, components and access interfaces; and/or, for providing access control for information interchange to and from the Intelligent Objects, Intelligent Object Pool 204, iPools 155, components and interfaces.

In an exemplary embodiment, an Intelligent Object-to-Intelligent Object Query interface **(OQMi) 456** contains a set of software instructions comprising meta-data indices definitions, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Intelligent Object-to-Intelligent Object Query component **(OQMc) 455**; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202** and **Intelligent Objects 200**. Methods and processes are provided, which detect and extract direct information required for object-to-object queries provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information; relationship information; protocols; ontologies; annotations and other data-enabling information; provided directly to the Intelligent Object-to-Intelligent Object Query component.

Additionally, methods and processes are provided, which route information derived from the Intelligent Object-to-Intelligent Object Query component directly to the **Intelligent Object Pool 204**; **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of the Intelligent Object-to-Intelligent Object Query component **(OQMc) 455** activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary aspect, the Intelligent Object-to-Intelligent Object Query interface **(OQMi) 456** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the Intelligent Object-to-Intelligent Object Query interface **(OQMi) 456** comprised as a module within an information technology platform using object-oriented data structures.

<u>iPool-to-iPool Query component (PPQc) 457</u>

The iPool-to-iPool Query component (PPQc) 457 is primarily responsible for query optimization based on provision of meta-data definitions for accessing, routing, and processing, of direct iPool-meta-data to iPool-meta-data information interchange, utilizing direct interchange

of linked meta-data index information; corresponding to automated and/or user-directed queries and/or processing requests.

In an exemplary embodiment, an iPool-to-iPool Query component **(PPQc) 457** contains a set of software instructions comprising meta-data indices methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Unified Presentation Layer; iPool-to-iPool Query interface___; and meta-data indices components. A set of meta-data indices definitions, which provide automated and/or interactive methods and processes including query optimization based on provision of meta-data definitions for accessing; routing and processing; of direct iPool meta-data-to-iPool meta-data information interchange; utilizing direct interchange of linked meta-data index information; corresponding to automated and/or user-directed queries and/or processing requests. A set of meta-data indices definitions, which provide automated and/or interactive methods and processes including IPool-to-IPool information interchange optimization based on organization of the IPool content and meta-data linking according to information including but not limited to content attributes; validation state; ranking; relationships; associations; and the like, and for query optimization based on dynamic updating of the IPool organization for information interchange according to correspondence between the meta-data information and queries; query histories; commands; command histories; and/or other access, routing and processing actions. A set of meta-data indices definitions, which provide automated and/or interactive methods and processes including query optimization based on user defined updating and management of the IPool organization for information interchange.

In an exemplary aspect, the iPool-to-iPool Query component **(PPQc) 457** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the iPool-to-iPool Query component **(PPQc) 457** comprised as a module within an information technology platform **151** using object-oriented data structures.

<u>iPool-to-iPool Query interface (PPQi) 458</u>

The iPool-to-iPool Query interface (PPQi) 458 is primarily responsible for detecting and extracting direct information desired or required for iPool-to-iPool queries provided by the Intelligent Objects 200 and their data content, Intelligent Object Pool 204, iPools 155, components, and access interfaces; where the information may include but not limited to one or more of meta-data indices including meta-data for content information, attribute information, relationship information, protocols, ontologies, annotations and other data-enabling information;

provided directly to the iPool-to-iPool Query component. It may also be responsible for routing information derived from the iPool-to-iPool Query component (PPQc) 457 directly to the Intelligent Object Pool (IOP) 204, iPools 155, components and access interfaces.

In an exemplary embodiment, an iPool-to-iPool Query interface **(PPQi) 458** contains a set of software instructions comprising meta-data indices definitions, methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following iPool-to-iPool Query component **(PPQc) 457**; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202** and **Intelligent Objects 200**. Methods and processes are provided, which detect and extract direct information required for iPool-to-iPool queries provided by the **Intelligent Objects 200** and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to meta-data indices; including meta-data for content information; attribute information; relationship information; protocols; ontologies; annotations and other data-enabling information; provided directly to the iPool-to-iPool Query component.

Additionally, methods and processes are provided, which route information derived from the iPool-to-iPool Query component **(PPQc) 457** directly to the Intelligent Object Pool **(IOP) 204**; **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the **Intelligent Objects 200**, **Intelligent Object Pool 204** , **iPools 155**, components and access interfaces. Methods and processes also include provision of the iPool-to-iPool Query component **(PPQc) 457** activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, the iPool-to-iPool Query interface **(PPQi) 458** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the iPool-to-iPool Query interface **(PPQi) 458** comprises an optional module within an information technology platform (ITP) **151** using object-oriented data structures **152**.

Other modules, components, and interfaces 459, 460, 461, 462, 463, 464 may optionally be provided to satisfy particular application, data handling, processing, interfacing, or other needs as the system and architecture, as well as the method are readily expandable.

Exemplary Pool Content Access Architecture, Methods and Protocols

The Pool Content Access architecture, methods and protocols are primarily responsible for providing content access and presentation definitions for significance detection, meta-definition aggregation, and result aggregation, results generation, and relationship viewing. It may also provide for distributed learning and knowledge extraction.

The Pool Content Access architecture and Pool Content Access methods defined herein provide content access and presentation definitions for significance detection, result aggregation and results generation. Pool Content Access protocols include modules comprising components and access interfaces for aggregate meta-definition, result aggregation and relationship viewing. In optional and preferred embodiments, the Pool Content Access Protocols may include modules for distributed learning and knowledge extraction; such as Distributed Learning Engines (DLEc/DLEi) **479, 480** and Distributed Extraction Engine interfaces **(DLEi) 480**; and/or Knowledge Extraction Engines (KEEc/KEEi) **481, 482** and Knowledge Extraction Engine interfaces **(KEEi) 482**.

Data Pool Content Access and Presentation Protocols 420

The Data Pool Content Access and presentation protocols 420 are primarily responsible for one or more of the following: providing meta-data index generation of aggregated Intelligent Object meta-data based on query parameters, significance detection of values located within the global data pool based on query parameters and global object meta-data index content, organizing meta-data index based object relationships within individual iPools to allow for real-time result aggregation and real-time exclusion of irrelevant object data layers, and optionally distributed learning and knowledge extraction.

In one embodiment of the inventive system, method, and computer program product, the Data Pool Content Access and presentation protocols, comprise: (a) an Aggregate Meta-data Index Generator which provides for meta-data index generation of aggregated Intelligent Object meta-data, based on query parameters; (b) an Aggregate Real-time Significance Generator which provides for significance detection of values located within the global data pool based on query parameters and global object meta-data index content; (c) an Object Property-Selective Pre-sorting Tool referred to as the IMO Zoom (IMO-Z), which organizes meta-data index based object relationships within individual iPool s to allow for real-time result aggregation and real-time exclusion of irrelevant object data layers; and (d) in optional and advantageous embodiments, additional modules that provide for distributed learning and knowledge extraction.

In an exemplary embodiment, a set of Pool Content Access definitions contains sets of software instructions comprising methods and processes, including bi-directional information interchange with components and access interfaces including but not limited to the following: an external **Intelligent Object Handler 202** (IOH); Intelligent Object Pools **(IOP) 204**; **iPools 155**; and Intelligent Objects or **Intelligent Molecular Objects 200**. The Pool Content Access definitions include automated and/or user-directed processes, methods and algorithms including but not limited to real-time significance detection; results aggregation; methods application; relationship viewing; and report generation utilizing results generated by the Pool Content Access definition modules including by object-to-analysis tools interactions; real-time significant answer generation; result merging algorithms and clustering algorithms; utilizing calculation methods not limited to Jacquard coefficient; Dice coefficient; Jeffrey coefficient; Pearson coefficient; simple matching; product moment correlation coefficient; mean square difference and absolute difference; and clustering methods not limited to minimum variance; single, complete, average and weighted average linking; median method; centroid method; neighbor joining method; Fitch-Margoliash least square method and Fitch-Margoliash evolutionary clock method.

In advantageous and optional embodiments, modules including one or more of learning and knowledge extraction algorithms are provided. In another embodiment, the Pool Content Access definitions are provided within an information technology platform (ITP) **151** using object-oriented data structures **152**.

Aggregate Meta-data Index Generator component (aMDXc) 471

The Aggregate Meta-data Index Generator component (aMDXc) 471 is primarily responsible for providing Pool Content Access enabling bi-directional information interchange with components and access interfaces, and providing automated and/or interactive methods and processes including generation of meta-data definitions for iPool data subsets.

In an exemplary embodiment, an Aggregate Meta-data Index Generator component **(aMDXc) 471** contains a set of software instructions comprising Pool Content Access enabling bi-directional information interchange with components and access interfaces including but not limited to Unified Presentation Layer; Aggregate Meta-data Index Generator interface **(aMDXi) 472**; and Pool Content Access components. Additionally, a set of Pool Content Access definitions are comprised which provide automated and/or interactive methods and processes including generation of meta-data definitions for iPool data subsets; including but not limited to automatically provided and/or user-defined information regarding Intelligent Object and iPool

data Intelligent Object data content accessed via vector pointers and meta-data index definitions; including meta-data for iPool subset attributes; annotations; protocols; ontologies; content attributes; data type, structure, and access dependencies; ownership and access information; relationship information; activity ranking; significance ranking and validation ranking.

For meta-data index generation, the information is organized, sorted and ranked according to comprised algorithms for determination of such as not limited to structure, function, validation, dependency, similarity; association; contiguity; proximity, weight and frequency, for such as query optimization requirements, based on correspondence of submitted query parameters; and query histories; to iPool Meta-data index content; and iPool access organization. Generation of iPool meta-data information required for methods and processes including but not limited to iPool activation; iPool sorting; iPool-to-iPool queries; iPool data presentation and result generation.

In an exemplary aspect, the Aggregate Meta-data Index Generator component **(aMDXc) 471** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the Aggregate Meta-data Index Generator component **(aMDXc) 471** comprised as a module within an information technology platform using object-oriented data structures.

<u>Aggregate Meta-data Index Generator interface (aMDXi) 472</u>

The Aggregate Meta-data Index Generator interface (aMDXi) 472 is primarily responsible for detecting and extracting direct information required for meta-data index generation, provided by the Intelligent Objects 200 and their data content; Intelligent Object Pool 204, iPools 155, components and access interfaces. It may also be responsible for routing information derived from the Aggregate Meta-data Index Generator component (aMDXc) 471 directly to the Intelligent Object 200, Intelligent Object Handler 202, Pool, and iPools 155, components and access interfaces.

In an exemplary embodiment, an Aggregate Meta-data Index Generator interface **(aMDXi) 472** contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces of and/or including but not limited to the following Aggregate Meta-data Index Generator component **(aMDXc) 471**; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202**; and **Intelligent Molecular Objects 200**. Methods and processes are provided, which detect and extract direct information required for meta-data index generation,

provided by the **Intelligent Objects 200** and their data content; **Intelligent Object Pool 204** ; **iPools 155**;components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information relationship information; protocols; ontologies; annotations; and other data-enabling information; provided directly to the Aggregate Meta-data Index Generator component **(aMDXc) 471.**

Additionally, methods and processes are provided, which route information derived from the Aggregate Meta-data Index Generator component **(aMDXc) 471**directly to the **Intelligent Object 200, Intelligent Object Handler 202** , Pool and **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes for ……………….. are provided, including but not limited to access control for information interchange to and from the **Intelligent Objects 200, Intelligent Object Pool 204, iPools 155**, components and interfaces. Comprised methods and processes also include provision of Aggregate Meta-data Index Generator component **(aMDXc) 471** activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary aspect, the Aggregate Meta-data Index Generator interface **(aMDXi) 472** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the Aggregate Meta-data Index Generator interface **(aMDXi) 472**comprised as a module within an information technology platform using object-oriented data structures.

Aggregate Real-time Significance Generator component (aRSGc) 473

The Aggregate Real-time Significance Generator (aRSGc) 473 is primarily responsible for the application of algorithms and procedures, such as applications and procedures that perform normalized comparison, generate correlation and significance detection information including but not limited to Intelligent Object properties, data content values, and meta-data index content; based on query parameters. It may also be responsible for organizing or enabling organization of Intelligent Object and iPools.

In an exemplary embodiment an Aggregate Real-time Significance Generator component **(aRSGc) 473** contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access

interfaces including but not limited to the following Unified Presentation Layer, Aggregate Real-time Significance Generator interface **(aRSGi) 474** and Pool Content Access components. Additionally a set of Pool Content Access definitions are included, which provide automated and/or interactive methods and processes including application of algorithms for normalized comparison; correlation and significance detection information including but not limited to Intelligent Object properties; data content values and meta-data index content; based on parameters fielded by all valid queries. The Aggregate Real-time Significance Generator enables organization of Intelligent Object(s) and **iPools**, utilizing methods and processes including assembly; ranking; classification; and/or tabulation of Intelligent Objects; according to comparison of normalized parameters such as data content values; and meta-data index properties provided by direct Intelligent Object-to-Intelligent Object; and iPool-to-iPool information interchange; via addressing including but not limited to vector subset pointing; meta-data index addressing; and activated by such as query submissions, user preference rules and/or viewing instructions.

The **Aggregate Real-time Significance Generator component (aRSGc) 473** enables organization of data utilizing methods and processes including assembly; ranking; classification; and/or tabulation of Intelligent Objects; based on the significance of results generated by external applications; components; access interfaces; and activated by query submissions, user preference rules and/or viewing instructions.

In an exemplary embodiment, the Aggregate Real-time Significance Generator component **(aRSGc) 473** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the Aggregate Real-time Significance Generator component optionally comprises a module within an information technology platform using object-oriented data structures.

<u>Aggregate Real-time Significance Generator interface (aRSGi) 474</u>

The Aggregate Real-time Significance Generator interface (aRSGi) 474 is primarily responsible for detecting and extracting direct information required for significance generation provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155, components and access interfaces. It may also be responsible for information derived from the Aggregate Real-time Significance Generator component (aRSGc) 473 directly to the Intelligent Object Pool 204, iPools 155, components, and access interfaces; and for access control for information interchange to and from the Intelligent Objects, Intelligent Object Pool 204, iPools

155, components and interfaces. It may further be responsible for aggregate real-time or near-real-time significance generation activity information to access interfaces and components as required for utility.

In an exemplary embodiment, an Aggregate Real-time Significance Generator interface **(aRSGi) 474** contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces of and/or including but not limited to the following Aggregate Real-time Significance Generator component **(aRSGc) 473**; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202**; and **Intelligent Molecular Objects 200** . Methods and processes are provided, which detect and extract direct information required for significance generation provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information relationship information; protocols; ontologies; annotations; and other data-enabling information; provided directly to the Aggregate Real-time Significance Generator component.

Additionally, methods and processes are provided, which route information derived from the Aggregate Real-time Significance Generator component **(aRSGc) 473** directly to the **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of Aggregate Real-time Significance Generator component **(aRSGc) 473** activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary aspect, the Aggregate Real-time Significance Generator interface **(aRSGi) 474** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the Aggregate Real-time Significance Generator interface **(aRSGi) 474** comprised as a module within an information technology platform using object-oriented data structures.

Result Aggregation Engine Component (RAEc) 224, 475

The Result Aggregation Engine Component (RAEc) 224, 475 is primarily responsible for aggregation of results provided by the aggregate significance generator module based on input or generated parameters and/or inquiries, such as for example queries, commands, processing requests, alerts, updates, and reports; such as may be presented by an object handler such as by Intelligent Object Handler 202. It may also be responsible for generating customized meta-data profiles, and/or generating aggregated meta-data outputs and updates such as iPool definition updates, iPool relationship definitions, Intelligent Object definition updates, and Intelligent Object relationship definitions pertaining to Intelligent Object and iPool properties and values provided by the Aggregate Significance Generator module. It may also be responsible for results assembly, validation, ranking, classification, and tabulation; for example, according to significance and relationship information provided by the Aggregate Significance Generator module.

In an exemplary embodiment, a Result Aggregation Engine component **(RAEc) 224, 475**, which contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Unified Presentation Layer; Result Aggregation Engine interface **(RAEi) 225, 476**, and Pool Content Access components. The Result Aggregation Engine Component is responsible for aggregation of results provided by the Aggregate Significance Generator module, according to parameters submitted via automated; and/or user-directed queries; commands; processing requests; alerts; updates; and reports; and presented via an external **Intelligent Object Handler 202**; and to generate customized meta-data profiles; and/or generate aggregated meta-data outputs and updates such as iPool definition updates; iPool relationship definitions; Intelligent Object definition updates; and Intelligent Object relationship definitions pertaining to Intelligent Object and iPool properties and values provided by the Aggregate Significance Generator module. A set of methods and processes, which provide results assembly, validation, ranking, classification, and tabulation; according to significance and relationship information provided by the Aggregate Significance Generator module.

In an exemplary embodiment, the Result Aggregation Engine component **(RAEc) 475** comprised as a module within an information technology platform for Intelligent Objects. In another embodiment, the Result Aggregation Engine component **(RAEc) 475** comprises a module within an information technology platform using object-oriented data structures.

<u>Result Aggregation Engine Interface (RAEi) 225, 476</u>

The Result Aggregation Engine Interface (RAEi) 225, 476 is primarily responsible for detecting and extracting direct information required for result aggregation, provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155, components, and access interfaces. It may also be responsible for routing information derived from the Result Aggregation Engine component (RAEc) 475 to the Intelligent Object, Intelligent Object Handler, Pool and iPools, and components and access interfaces. It may further be responsible for access control for information interchange to and from the Intelligent Objects, Intelligent Object Pool 204, iPools 155, components and interfaces.

In an exemplary embodiment, a Result Aggregation Engine interface **(RAEi) 476** contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces of and/or including but not limited to the following Result Aggregation Engine component **(RAEc) 475**; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202**; and **Intelligent Molecular Objects 200**. Methods and processes are provided, which detect and extract direct information required for result aggregation, provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information relationship information; protocols; ontologies; annotations; and other data-enabling information; provided directly to the **Result Aggregation Engine component (RAEc) 475.** Additionally, methods and processes are provided, which route information derived from the **Result Aggregation Engine component (RAEc) 475** directly to the Intelligent Object, **Intelligent Object Handler 202**, Pool and **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of **Result Aggregation Engine component (RAEc) 224,475** activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, the **Result Aggregation Engine interface (RAEi) 225, 476** comprised as a module within an information technology platform for Intelligent Objects. In an optional aspect, the **Result Aggregation Engine interface (RAEi) 476** optionally comprises a module within an information technology platform using object-oriented data structures.

IMO Zoom component (IMO-Zc) 477

The IMO Zoom component (IMO-Zc) 477 is primarily responsible for access, extraction, processing and addressing of information including of graphical viewing information; including but not limited to data relationship information and overviews, self-organizing maps, similarity clustering, dendrograms, charts, tables, and/or additional graphical representations, and combinations thereof. It may also be responsible for enabling thumbnail viewing of Intelligent Object data content, iPool data content, according to correspondence of data content attributes to automated and/or user-directed queries and/or commands. It may further be responsible for enabling graphical pre-sorting and sorting of Intelligent Object data content, and iPool data content, according to correspondence of data content attributes to automated and/or user-directed queries and/or commands.

An IMO Zoom component **(IMO-Zc) 477** contains a set of software instructions comprising Pool Content Access methods and processes including but not limited to bi-directional information interchange with components and access interfaces including but not limited to the following Unified Presentation Layer; IMO Zoom interface **(IMO-Zi) 478**; and Pool Content Access components. Comprised methods and processes also include access; extraction; processing and addressing of graphical viewing information; including but not limited to data relationship information and overviews; self-organizing maps, similarity clustering; dendrograms; charts; tables; and/or additional graphical representations. Additionally, a set of instructions enable thumbnail viewing of such as Intelligent Object data content; iPool data content; according to correspondence of data content attributes to automated and/or user-directed queries and/or commands. Additionally, a set of instructions enable user-directed and/or automated graphical pre-sorting and sorting of such as the Intelligent Object data content; iPool data content; according to correspondence of data content attributes to automated and/or user-directed queries and/or commands. The information being accessed from and routed to the Intelligent Object data; **Intelligent Object Pool 204** data; iPool data; modules, components, and access interfaces.

Example of Code for Clustering of Intelligent Object Data within an IOP

The following example of computer program software code for clustering of Intelligent Object data within the **Intelligent Object Pool 204** for viewing and analysis utilizes content attributes and attribute flags of the corresponding objects. The Cluster Dialog box, which is evoked below, allows for selection of the adequate calculation method for the similarity coefficient (e.g. Jacquard, Dice, Jeffrey, Pearson, and/or other algorithms or procedures as are known in the art), the parameter the clustering is performed upon (e.g. spot intensity, spot volume, spot color, spot position, band position, band concentration, band molecular weight, band iso-electric point, and the like), and the method for clustering applied to display numerically and/or graphically the data relationships (such as Minimum Variance [MVAR], Minimum Variance Hierarchy [MNVR], Single Link [SLNK], Complete Link [CLNK], Average Link [ALNK], Weighted Average Link [WLNK], Median Method, Centroid Method, Minimum Spanning Tree [MST], Neighbor-Joining Method, FMLS, FMLS with Evolutionary Clock and the like). User preferred clustering parameters can be saved and reloaded to be applied to similar data objects consequently.

Table I. Exemplary computer program code for performing the procedure

Table I. Exemplary computer program code for performing an embodiment of the procedure.

```
// CClusterDlg dialog

class CClusterDlg : public Cdialog
{
// Construction
public:
        CClusterDlg(CWnd* pParent = NULL);   // standard constructor

// Dialog Data
        //{{AFX_DATA(CClusterDlg)
        enum { IDD = IDD_CLUSTER };
        int             m_nCalibMethod;
        //}}AFX_DATA


// Overrides
        // ClassWizard generated virtual function overrides
        //{{AFX_VIRTUAL(CClusterDlg)
        protected:
        virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
        //}}AFX_VIRTUAL
```

```
// Implementation
protected:

        // Generated message map functions
        //{{AFX_MSG(CClusterDlg)
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};


// CClusterDlg dialog


CClusterDlg::CclusterDlg(CWnd* pParent /*=NULL*/)
        : CDialog(CClusterDlg::IDD, pParent)
{
        //{{AFX_DATA_INIT(CclusterDlg)
        m_nCalibMethod = 0;
        //}}AFX_DATA_INIT
}


void CClusterDlg::DoDataExchange(CDataExchange* pDX)
{
        //{{AFX_DATA_MAP(CclusterDlg)
        DDX_CBIndex(pDX, IDC_CALIBMETHOD, m_nCalibMethod);
        //}}AFX_DATA_MAP
}


BEGIN_MESSAGE_MAP(CClusterDlg, Cdialog)
        //{{AFX_MSG_MAP(CClusterDlg)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()

...

BEGIN_MESSAGE_MAP(CcontentAttribDlg, CDialog)
        //{{AFX_MSG_MAP(CcontentAttribDlg)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()


CContentAttribDlg::CcontentAttribDlg(CWnd* pParent /*=NULL*/)
        : CDialog(CcontentAttribDlg::IDD, pParent)
{
        //{{AFX_DATA_INIT(CcontentAttribDlg)
                // NOTE: the ClassWizard will add member initialization here
```

```
        //}}AFX_DATA_INIT
}


void CcontentAttribDlg::DoDataExchange(CDataExchange* pDX)
{
  //{{AFX_DATA_MAP(CcontentAttribDlg)
  DDX_Control(pDX, IDC_TREE1, m_treeContentAttrib);
  //}}AFX_DATA_MAP
}


BOOL CcontentAttribDlg::OnInitDialog()
{
        CDialog::OnInitDialog();

  FillTree();

  return TRUE;  // return TRUE unless you set the focus to a control
                // EXCEPTION: OCX Property Pages should return FALSE
}


void CcontentAttribDlg::FillTree()
{
  // create the tree
  BYTE   flag = 0;
  CString str = _T("");
  HTREEITEM parent = TVI_ROOT;
  for(int i = 0; i < 1000; i++)
  {
    GetLutContentAttrib(i, str, flag);
    if(!str.IsEmpty())
    {
      if(flag & 1)
      {
        parent = m_treeContentAttrib.InsertItem(str, parent);

        WORD l  = flag;
        WORD h  = (WORD)i;
        DWORD f = MAKELONG(l, h);
        m_treeContentAttrib.SetItemData(parent, f);
      }

      if(flag & 2)
      {
        CString fmt = _T("");
        fmt.Format(_T("(%.3d) - %s"), i, str);
        HTREEITEM item = m_treeContentAttrib.InsertItem(fmt, parent);
```

```
        WORD l  = flag;
        WORD h  = (WORD)i;
        DWORD f = MAKELONG(l, h);
        m_treeContentAttrib.SetItemData(item, f);
      }

      if(flag & 4)
        parent = m_treeContentAttrib.GetParentItem(parent);

      if(flag & 8)
      {
        parent = m_treeContentAttrib.GetParentItem(parent);
        parent = m_treeContentAttrib.GetParentItem(parent);
      }

      if(flag & 16)
      {
        parent = m_treeContentAttrib.GetParentItem(parent);
        parent = m_treeContentAttrib.GetParentItem(parent);
        parent = m_treeContentAttrib.GetParentItem(parent);
      }

      if(flag & 32)
      {
        parent = m_treeContentAttrib.GetParentItem(parent);
        parent = m_treeContentAttrib.GetParentItem(parent);
        parent = m_treeContentAttrib.GetParentItem(parent);
        parent = m_treeContentAttrib.GetParentItem(parent);
      }
    }
  }
}
(End Code Example)
```

In an exemplary embodiment, the IMO Zoom component **(IMO-Zc) 477** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the IMO Zoom component **(IMO-Zc) 477** optionally comprises a module within an information technology platform using object-oriented data structures.

IMO Zoom interface (IMO-Zi) 478

The IMO Zoom interface (IMO-Zi) 478 is primarily responsible for detection of graphical content provided by the Intelligent Objects 200 and their data content, Intelligent Object

Pool 204, iPools 155, components and access interfaces; accessing and routing of the information via such means as vectorized data field pointers, and meta-data index definitions, and provided by the IMO Zoom component. It may also be responsible for routing graphical information derived from the IMO Zoom component (IMO-Zc) 477 to the Intelligent Object Pool 204, iPools 155, components, and access interfaces (interfaces or module/component interfaces). It may further be responsible for providing and/or enabling information interchange to and from the Intelligent Objects, Intelligent Object Pool 204, iPools 155, components and interfaces.

An IMO Zoom interface **(IMO-Zi) 478** contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following IMO Zoom component **(IMO-Zc) 477**; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202** ; and **Intelligent Molecular Objects 200**. Methods and processes are provided comprising detection of graphical content provided by the **Intelligent Objects 200** and their data content; **Intelligent Object Pool 204** ; **iPools 155**; components; and access interfaces; accessing and routing of the information via such as vectorized data field pointers; and meta-data index definitions; and provided by the IMO Zoom component 477.

Additionally, methods and processes are provided, which route graphical information derived from the IMO Zoom component **(IMO-Zc) 477** to the **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of IMO Zoom component **(IMO-Zc) 477** activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, the IMO Zoom interface **(IMO-Zi) 478** comprises a module within an information technology platform **151** for Intelligent Objects. In another embodiment, the IMO Zoom interface **(IMO-Zi) 478** comprises a module within an information technology platform using object-oriented data structures.

Distributed Learning Engine Component (DLEc) 479

The Distributed Learning Engine Component (DLEc) 479, when optionally present, is primarily responsible for knowledge extraction and is optionally provided in the inventive system, method, and computer program.

In one embodiment, a Distributed Learning Engine component **(DLEc) 479** contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces including but not limited to the following Unified Presentation Layer, Distributed Learning Engine interface **(DLEi) 480** and Pool Content Access components. Methods and processes for knowledge extraction utilizing automated and/or user-directed interactive application of comprised protocols, functions and algorithms to information and actions including but not limited to user queries; automated queries; customized sets of queries; data objects such as the Intelligent Object; data handlers such as the **Intelligent Object Handler 202**; data pools such as the **Intelligent Object Pool 204**; **iPools 155**; the global data entity; and/or data subsets therein. Methods and processes are provided for the automation of learning processes including but not limited to simulation; prediction, hypothesis generation, testing, result assessment and comparison; interactive result optimization and feedback; data and user workspace definition; assessment; organization; customization; data and user workspace relationship definition; assessment; organization; customization; and/or optimization of accessing, routing and processing protocols for user workspaces; Intelligent Objects; Intelligent Object Pools and **iPools 155**; utilizing a comprised set of Pool Content Access definitions, which may include but are not limited to Cross-reference clustering; Boolean network algorithms; Bayesian network algorithms; neural network algorithms; iterative learning algorithms; time series analysis algorithms; pattern matching algorithms; structure matching algorithms; and rule induction algorithms. Methods are provided for the application of the algorithms to the information to enable such as automated, dynamic and/or interactive iterative processing; significance detection; tabulation; validation; ranking; assembly; and/or other forms of distributed learning.

In another embodiment, the Distributed Learning Engine component **(DLEc) 479** comprised as a module within an information technology platform for Intelligent Objects. In another embodiment, the Distributed Learning Engine component **(DLEc) 479** optionally comprises a module within an information technology platform using object-oriented data structures.

Distributed Learning Engine Interface (DLEi) 480

The Distributed Learning Engine interface (DLEi) 480, when optionally present, is primarily responsible for detecting and extracting direct information required for distributed learning, provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155, components and access interfaces, and is optionally provided in the inventive system, method, and computer program.

In an optional embodiment, a Distributed Learning Engine interface **(DLEi) 480** contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces of and/or including but not limited to the following Distributed Learning Engine component **(DLEc) 479**; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202**; and **Intelligent Molecular Objects 200**. Methods and processes are provided, which detect and extract direct information required for distributed learning, provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information relationship information; protocols; ontologies; annotations; and other data-enabling information; provided directly to the Distributed Learning Engine component **(DLEc) 479**. Additionally, methods and processes are provided, which route information derived from the Distributed Learning Engine component **(DLEc) 479** directly to the Intelligent Object, **Intelligent Object Handler 202**, Pool and **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204**, **iPools 155**, components and interfaces. Comprised methods and processes also include provision of Distributed Learning Engine component **(DLEc) 479** activity information to access interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, the Distributed Learning Engine interface **(DLEi) 480** comprised as a module within an information technology platform for Intelligent Objects. In another exemplary embodiment, the Distributed Learning Engine interface comprises a module within an information technology platform optionally using object-oriented data structures.

Knowledge Extraction Engine component (KEEc) 481

The Knowledge Extraction Engine component (KEEc) 481, when optionally present, is primarily responsible for knowledge extraction utilizing automated and/or user-directed application of protocols, functions and algorithms to information and actions including but not limited to user queries, automated queries, customized sets of queries, data objects such as the Intelligent Object, data handlers such as the Intelligent Object Handler 202, data pools such as the Intelligent Object Pool 204, iPools 155, the global data entity, and/or data subsets therein.

In an optional embodiment, a Knowledge Extraction Engine component **(KEEc) 481**, which contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces including but not limited to Unified Presentation Layer; Knowledge Extraction Engine interface **(KEEi) 482**; and Pool Content Access components. Methods and processes for knowledge extraction utilizing automated and/or user-directed interactive application of comprised protocols, functions and algorithms to information and actions including but not limited to user queries; automated queries; customized sets of queries; data objects such as the Intelligent Object; data handlers such as the **Intelligent Object Handler 202**; data pools such as the **Intelligent Object Pool 204**; **iPools 155**; the global data entity; and/or data subsets therein. Methods and processes are provided for the automation of knowledge extraction processes including but not limited to active data attribute searching, detection and extraction; similarity and significance assessment and comparison; hypothesis generation; report generation; interactive result optimization and feedback; data annotation and user workspace definition, assessment, organization and customization and/or optimization; data annotation and user workspace relationship definition, assessment, organization and customization and/or optimization of accessing, routing and processing protocols for user workspaces; Intelligent Objects; Intelligent Object Pools and **iPools 155**; utilizing a comprised set of Pool Content Access definitions, which may include but are not limited to the following Cross-reference clustering; Boolean network algorithms; Bayesian network algorithms; neural network algorithms; iterative knowledge extraction algorithms; time series analysis algorithms; pattern matching algorithms; structure matching algorithms; and rule induction algorithms. Methods are provided for the application of the algorithms to the information to enable such as automated, dynamic and/or interactive iterative processing; significance detection; tabulation; validation; ranking; assembly; and/or other forms of knowledge extraction.

In an exemplary embodiment, the Knowledge Extraction Engine component **(KEEc) 481** comprised as a module within an information technology platform for Intelligent Objects. In an optional embodiment, the Knowledge Extraction Engine component **(KEEc) 481** comprised as a module within an information technology platform using object-oriented data structures.

<u>Knowledge Extraction Engine interface (KEEi) 482</u>

The Knowledge Extraction Engine interface (KEEi) 482, when optionally present, is primarily responsible for detecting and extracting direct information required for knowledge extraction, provided by the Intelligent Objects and their data content, Intelligent Object Pool 204, iPools 155, components, and access interfaces; . provided directly to the Knowledge Extraction Engine component (KEEc) 481. The information may include one or more of but not limited to Intelligent Object data content accessed via vector pointers, meta-data indices, including meta-data for content information, attribute information relationship information, protocols, ontologies, annotations, and other data-enabling information.

In an optional embodiment, a Knowledge Extraction Engine interface **(KEEi) 482** contains a set of software instructions comprising Pool Content Access methods and processes including bi-directional information interchange with components and access interfaces of and/or including but not limited to the following Knowledge Extraction Engine component (KEEc) **481**; Intelligent Object Pool access interfaces; **Intelligent Object Handler 202**; and **Intelligent Molecular Objects 200**. Methods and processes are provided, which detect and extract direct information required for knowledge extraction, provided by the Intelligent Objects and their data content; **Intelligent Object Pool 204**; **iPools 155**; components; and access interfaces; the information including but not limited to Intelligent Object data content accessed via vector pointers; meta-data indices; including meta-data for content information; attribute information relationship information; protocols; ontologies; annotations; and other data-enabling information; provided directly to the Knowledge Extraction Engine component **(KEEc) 481.**

Additionally, methods and processes are provided, which route information derived from the Knowledge Extraction Engine component **(KEEc) 481**directly to the Intelligent Object, **Intelligent Object Handler 202**, Pool and **iPools 155**; components and access interfaces. Automated and/or interactive methods and processes are provided, including but not limited to access control for information interchange to and from the Intelligent Objects, **Intelligent Object Pool 204, iPools 155**, components and interfaces. Comprised methods and processes also include provision of Knowledge Extraction Engine component **(KEEc) 481** activity information to access

interfaces and components as required for utility including but not limited to synchronization of the component information interchange and activity recordation of the component information interchange and activity; auditing of the component information interchange and activity; validation of the component information interchange and activity.

In an exemplary embodiment, the Knowledge Extraction Engine interface **(KEEi) 482** comprises a module within an information technology platform for Intelligent Objects. In another embodiment, the Knowledge Extraction Engine interface **(KEEi) 482** comprises a module within an information technology platform optionally using object-oriented data structures.

Through provision of the architecture, methods, and modules, the infrastructure is provided for secure, unified object storage and object-to-object and iPool-to-iPool query-based interaction, to allow for a comprehensive real-time analysis of otherwise inaccessible, inconsistent data sets. It is evident from the above description that the data pool architecture and methods described above allow for efficient real-time processing of terabytes of complex, multidimensional, interdependent data, thereby providing real-time answers to queries within heterogeneous data environments with high data density and dynamic application needs.

Embodiments of the **Intelligent Object Pool 204** implemented as computer software program code may be compiled to run on any one or more of multiple platforms and/or operating system environments known in the art as well as those reasonably expected to develop in the future, including, but not limited to, UNIX and UNIX-based platforms and/or environments, Linux and Linux-based platforms and/or environment, Macintosh operating systems (such as OS 9 and OS 10) and Macintosh-based based platforms and/or environments, or any Windows platform and/or Windows-based platforms such as Microsoft Windows 98, ME, 2000, NT, XP and extensions and modifications thereof, and/or environment such as any Window 32-bit operating system, platform, or environment.

The hardware and system descriptions provided herein are intended define and describe the requirements for a specific exemplary embodiment, implemented for a 32-bit Microsoft Windows environment. However, it will be appreciated that a broad class of general purpose computing systems may be utilized and that the inventive methods, procedures, and architectures are by-and-large hardware, operating system, and network or connectivity agnostic.

With reference to illustration in FIG. 9, in a global, heterogeneous environment, data from diversified sources are functionally integrated via Intelligent Objects and their **Intelligent Object Pool 204**, utilizing automated access definition protocols, multiple alias integration and

addressing, data field mapping and state management to allow for integrated data access, analysis and exchange utilizing distributed data content presented within the Client iPool.

The chart depicts user relationships to the data on database and iPool levels. In **FIG. 9**, the upper section of the diagram (the section that includes the labels "Client iPool", "Restricted", "Client Data", "Resources", and the like) depicts data within intranets, LANs and the like, while the lower, larger part of the diagram contains several different forms of public and/or otherwise web-accessible data sources. Other examples of data resources are depicted in the diagram and are merely exemplary and not limiting in any way.

With reference to FIG. 10, it represents another exemplary embodiment of a process model describing interactions of required or included modules for an exemplary embodiment of the **Intelligent Object Pool 204**.

In this exemplary process model, data from a global data resource (or any part of a global data resource) are passed through an access interface, in this example an access interface having a security layer and a set of access and/or exchange protocols and integrity assessment procedures, to the Intelligent Objects. Lastly, the pathways involved in unified direct Intelligent-Object-to-Intelligent Object and iPool-to-iPool data interaction generate the real-time answer and pass it back through security to the **Intelligent Object Handler 202**.

By real time answer we generally mean an answer or results are received within a relatively short period of time so that a user inputting a query may wait for a response to the query and not find the wait objectionable to be contrasted with a batch processed query or other query that has significant waiting time. In many instances the results will be generated in less than a second so that there will be no apparent wait; however, for some queries the delay in receiving a response with results may be a few seconds, several seconds, and perhaps several minutes. The invention is not limited to any particular response time.

With reference to FIG. 11, there is illustrated a representation of the graphical user interface window and display screen image within the unified presentation layer displaying algorithms included for clustering of Intelligent Object data. In this example, a number of property panes showing dendrograms results are seen behind the clustering algorithms menu.

In this example, and in some of the examples and screen displays that follow, data from a local subset iPool of Intelligent Objects were queried against specific protein expressions based on 2-dimensional gel electrophoresis (2DE) data. During the real-time answer-finding process, object-to-object interactions are represented via a graphical iPool Viewer (IMO Zoom), which

also accesses relevancy of individual result contributions to generate a unique, exact, relevant real-time answer.

With reference to FIG. 12, there is illustrated an example of an interface iPool Zoom viewer and displayed image for viewing iPool data relationships, utilizing implemented techniques, including for example techniques that include one or more of dendrograms and self-organizing maps (SOM). Attributes and interactions between individual Intelligent Object (IMO) data within the set of objects are outlined in the margins of this exemplary display.

With reference to FIG. 13, there is illustrated an example of an iPool Zoom interface for viewing iPool data relationships, utilizing Principal Components Analysis (PCA).

Operating and Business Model Providing Information Services

Among its inventive aspects, the invention provides a revolutionary new information technology platform that places the power (in terms of response time to complex queries and analytical requests utilizing current tools, such as a common query in bioinformatics for detection of spots in a 2-D Electrophoresis gel) of an entire floor of clustered servers or 'massively parallel mainframe' computers (e.g. IBM, COMPAQ, or the like) at the hands of any scientist or consumer, for that matter, with a computer and a connection to the web. As a result, information processing, management and storage in every field imaginable (Life Sciences, Agribusiness, Large Scale Manufacturing, Physics Imaging, and many more) are dramatically and revolutionarily more efficient and cost effective.

The inventive system, method, and business model is projected toward an initial market in the Life Sciences industries as a result of internal expertise and a tremendous and growing need for the kind of time efficiency and cost effectiveness that our IT Platform will provide.

To further facilitate the need of Life Science and Life Scientist, the invention provides developments and advances toward a number of product modules (typically implemented as computer program software for execution on computer systems) ranging from Drug Discovery, Genomics, Proteomics to Metabolism product modules, that will reside upon the platform and thereby enable the dramatic shortening of timelines for new drug development and gene therapies (while also providing for rapidly, validated diagnosis and treatment) in a Real-Time and cost effective manner. The abbreviated timelines will, in turn, provide cost savings for each new drug to the tune of at least $200 million dollars, while facilitating sales, based on earlier than expected market entry, upwards of $2 billion, for each new drug.

These IT platforms will become a standard, particularly as purveyors of solutions for substantially improved drug discovery, via its various Drug Discovery, Genomic and Proteomic product modules.

The motivation for the platform architecture and its associated methods and procedures has arisen largely because, as a result of the human genome project and other related activities (genomics and proteomics), Biotech and Pharmaceutical companies are drowning in a flood of information, information which may hold the key to powerful and valuable new drug discoveries and gene therapies.

Currently, Biotechnology and Pharmaceutical companies are spending upwards of $40 billion each year to sift through this information in order to uncover new drug candidates and potential gene therapies. Despite the vast sums of money being spent, the task of finding new drug candidates and gene therapies remains daunting, costly and highly inefficient.

Some reasons for this can be traced to a number of factors, several of which include: a variety of different types and kinds of databases; applications and systems that cannot communicate with one another; the enormous cost to retool a company's existing information technology platform; the scarcity of bioinformatics specialist; and, the lack of appropriate analysis tools.

As a result, Biotechnology and Pharmaceutical companies are facing three critical issues. The pressure to reduce cost, to speed up the entire process for new drug development and to recover R&D cost more quickly via the sales of new drugs. To date, this remains a wholly unrealized goal.

As a part of the solution, the inventive IMO IT platform provides Biotech and pharmaceutical companies with the ability to quickly and cost effectively sort through the growing mass of information to discover and produce drugs in vastly reduced time periods and at greatly reduced cost. It will be possible as a result of the IMO™ IT platform and drug discovery, proteomics and genomics modules for Biotech and Pharmaceutical companies to shorten the drug discovery process by 2 to 4 years and save upwards of $200 million. Additionally, the various Biotech and Pharmaceutical companies will benefit from earlier than expected revenues (several billion dollars), as a result of reduced development time and thus earlier than anticipated market entry for each new drug.

The inventive system and methods therefore also provide or support a number of new and novel business model and operating model innovations that satisfy the needs of the information

community as well as provide revenue. Forming corporate strategic relationships are part of this overall concept.

Business development efforts related to the inventive technology include marketing the inventive products and services to Biotech and Pharmaceutical research companies initially, and to Life Science companies, in general.

Heretofore, the major participants in the field of Biotechnology software for data analysis have comprised the following three categories: Legacy Data Warehouses, data marts, ERP data mining tool companies, which provide proprietary applications and databases; Applications Service Providers (ASPs), Portals and other web-enabled service providers; and Network Integration Providers, which provide network integration of public databases, proprietary data and applications as well as support for local/remote collaboration and decision-making.

These established companies, have been, in general, committed to legacy software, narrowly useful web-based technology, or piecemeal component-based integration solutions, which have depended and continue to depend on expensive mainframe, server cluster, and hardware-enabled "parallel processing" computing to provide their analytical product.

As of yet, no clear leader has emerged to meet the demand for innovative software solutions within this rapidly expanding field, and therefore there remains a need for a more innovative and satisfactory solution.

Embodiments of the inventive system, method, and business model will generate revenue from, for example, at least one or more of the following areas; (1) the sale and licensing of its IT Platform, (2) the sale of its various Drug Development, Genomics, Cheminformatics and Proteomics modules, (3) the sale and licensing of its data-pool assets, (4) royalties from strategic collaborations, and (5) internal use of the IT Platform for production of valued information such as for internal drug discovery or monitoring of such as public health data. Other revenue streams are also contemplated.

Several exemplary application areas are now described by way of example. While the above referenced related patent applications have described innovations in information technology, especially for processing of high numbers of heterogeneous high-density data in heterogeneous computing environments, and more particularly in biotechnology, pharmaceutical, chemical, and life science environments, the invention is not so limited. The systems, methods, interfaces, engines, procedures, functions, algorithms, and other aspects of the invention as described here and in the related applications that are incorporated by reference may advantageously be applied to and/or used in conjunction with information systems generally,

physics imaging and analysis, intelligence integration and analysis, large scale manufacturing, agriculture and agribusiness, geographic information systems (GIS), the food industry, epidemiology, large scale forensics, economics and financial systems, health and human services, medical systems, as well as many other fields in which large amounts of data are involved.

In the field of Information Systems, applications of the inventive structure and method include but are not limited to Information Technology (IT) Platform(s), B2A infrastructure, database technology, and platform back-ends, among others. Some of the value in this area includes but is not limited to Flexible, Efficient, and Scalable Systems Integration; Data-enabling for Fast and Secure Data Access and Management; and Scalable and Efficient Applications Development Environment. Computer, network, and information systems providers may benefit from aspects of the invention.

In the field of Physics Imaging and Analysis, applications of the inventive structure and method include but are not limited to Groundwater, Oil, Mineral Exploration, Mining, Mapping, and Real-time Analysis. Some of the value in this area includes but is not limited to Added Efficiency and Functionality for Remote, Magnetic and Sonic Imaging and Analysis, Reduced Exploration Costs, and Increased Predictive Accuracy for Reduced Extraction Footprint. Organizations such as NASA, the Department of Energy, an mineral and resource exploration organizations may benefit from aspects of the invention.

In the field of Large Scale Manufacturing, applications of the inventive structure and method include but are not limited to Just-in-time (JIT) Inventory Management, Process Management, Robotics, and CAD/CAM. Some of the value in this area includes but is not limited to Improved Market, Acquisition and Inventory based on Global Data Access, Flexible and Scalable Process and Infrastructure Management, Real-time, and Integrated Process Optimization. Automobile manufacturers, chemical manufacturers, semiconductor manufacturers, and other large scale material and manufacturing organizations may benefit from the inventive technology.

In the field of Agribusiness, applications of the inventive structure and method include but are not limited to GMO's, Crop Engineering, Seed Banks and Animal Breeding. Some of the value in this area includes but is not limited to Enhanced Bioengineering Applications, Automated QA/QC, Integrated GLP/GMP, Inventory and Process Flow Automation, and Real-time Supply Chain Management. Chemical, textile, and other food research and production organizations may benefit from the inventive technology in this area.

In the Food Industry, applications of the inventive structure and method include but are not limited to Modified Additives, Food Instantization, Food and Foodstuffs processing, Manufacturing Process Design and Automation, and Inventory and Product Distribution. Some of the value in this area includes but is not limited to Enhanced Bioengineering Applications, Automated QA/QC, Integrated GLP/GMP, Inventory and Process Flow Automation, and Real-time Supply Chain Management, among others. Consumer food producers, processors, and packagers will benefit from such technology.

In the field of Epidemiology, applications of the inventive structure and method include but are not limited to Disease Studies, toxicology studies and analysis, and disease Outbreak Prevention. Some of the value in this area includes but is not limited to its Real-time capabilities, and its ability to provide Predictive Modules for Multidimensional Disease Studies and Diagnostics. For example, the Center for Disease control (CDC), the Department of Health and Human Services (DHHS), and various governmental and environmental laboratories may benefit from such technology.

In the field of forensics, particularly Large Scale Forensics, applications of the inventive structure and method include but are not limited to Fingerprint, DNA, and Materials Analysis, and Real-time Data Integration and Access. Some of the value in this area includes but is not limited to Real-time Access to Global Data Records, On-site Fingerprint, Photo Searching, and DNA matching. Law enforcement agencies such as the FBI, Interpol, and other investigative and law enforcement agencies will benefit from the technology, and in addition such organizations such as insurance companies and health maintenance organizations will benefit.

Therefore it will be appreciated that the invention is not limited to any particular field or application; rather, aspects of the invention may be applied to information technology generally where large amounts of heterogeneous data or information are involved.

Although the foregoing invention has been described in some detail by way of illustration and example for purposes of clarity of understanding, it will be readily apparent to those of ordinary skill in the art in light of the teachings of this invention that certain changes and modifications may be made thereto without departing from the spirit or scope of the appended claims.